Argonne
NATIONAL LABORATORY

ANL/NE-16/23

FY16 Status Report on NEAMS Neutronics Activities

Nuclear Engineering Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL/NE-16/23

FY16 Status Report on NEAMS Neutronics Activities

prepared by
C. H. Lee, E. R. Shemon, M. A. Smith, and Y. S. Jung
Nuclear Engineering Division, Argonne National Laboratory

September 30, 2016

EXECUTIVE ABSTRACT

The goal of the NEAMS neutronics effort is to develop a neutronics toolkit for use on sodium-cooled fast reactors (SFRs) which can be extended to other reactor types. The neutronics toolkit includes the high-fidelity deterministic neutron transport code PROTEUS and many supporting tools such as a cross section generation code MC^2-3, a cross section library generation code, alternative cross section generation tools, mesh generation and conversion utilities, and an automated regression test tool.

The FY16 effort for NEAMS neutronics focused on supporting the release of the SHARP toolkit and existing and new users, continuing to develop PROTEUS functions necessary for performance improvement as well as the SHARP release, verifying PROTEUS against available existing benchmark problems, and developing new benchmark problems as needed. The FY16 research effort was focused on further updates of PROTEUS-SN and PROTEUS-MOCEX and cross section generation capabilities as needed.

The PROTEUS-SN work was primarily focused on fast spectrum reactor analysis needs. In order to support the SHARP toolkit release, the PROTEUS-SN solver was updated to perform kinetics calculations driven by multi-physics coupled calculations within SHARP. Additionally, the mesh deformation simulations were streamlined such that the material density and isotopic (sodium) concentration updates are now performed consistently inside of PROTEUS-SN. This was handled with the existing material model input and assumed updates from the Diablo code on the mesh vertices. It is important to note that a major bug was identified and fixed in the coupled PROTEUS-SN coding which compromised the accuracy of some of the previous work. The new standalone version of PROTEUS-SN was released at the same time as the SHARP release in March 2016. Additional support and development on PROTEUS-SN has been performed based on requests by users.

As part of the verification process of SHARP, PROTEUS-SN was setup to automatically analyze deformed mesh simulations using the same coding implemented for the SHARP calculations. The new mesh deformation capability was verified against legacy procedures to compute reactivity feedback for a number of assumed deformed configurations for the ABTR design. Interestingly, the legacy procedures were shown to exaggerate the reactivity feedback due to radial core expansion by 35% compared to direct simulations by PROTEUS. This provides motivation for further work to show the real Diablo or NUBOW predicted deformation using a high fidelity code like PROTEUS-SN which the legacy procedures cannot predict.

The PROTEUS-MOCEX development was focused on two simultaneous tasks. First, the MOCEX methodology was implemented in a test code as part of the continued improvement to MC^2-3. This capability was needed for the Russian BFS experiment modeling and desired for more general SFR reactor analysis needs. A parallel capability was added to the test code

to make it more practical. More importantly, research into a CMFD acceleration of the MOCEX methodology was successfully demonstrated. When implemented, this will have a considerable impact on the performance of the PROTEUS-MOCEX code.

The PROTEUS-MOCEX specific improvements were focused on improving a ray tracing scheme and the replacement of the multi-group GMRES with a standard Gauss-Seidel scheme. The GMRES algorithm was maintained for the solving the diagonal inversion (within-group system) required in the Gauss-Seidel scheme. The Gauss-Seidel scheme is shown to reduce the computational effort and memory considerably.

Finally, considerable work was done on cross section generation techniques. For continued improvements in fast spectrum analysis work, a much needed thermal cross section treatment was added to MC^2-3. Preliminary analysis efforts indicate that this approach improves the DIF3D predicted results with measured results in EBR-II although that work is not discussed in this report. In addition to this improvement in MC^2-3, the gamma library needed for fast spectrum analysis was updated and is shown to provide results in good comparison with MCNP6. Finally, the cross section generation methodologies using the Serpent Monte Carlo code were extended to use the OpenMC Monte Carlo code. Combined with MC^2-3 and the cross section API, the Monte Carlo approach provides the user with multiple options for their analysis needs.

Overall, the preceding work demonstrates an ongoing commitment to improve the PROTEUS neutronics component of the NEAMS SHARP toolkit. In addition to the work completed on the topics above, this report shows additional verification and validation efforts on two thermal reactor problems: TREAT and Reactor Critical Facility (RCF). These two reactor systems were modeled and simulated using PROTEUS-MOCEX and demonstrated good agreement in eigenvalue with Serpent Monte Carlo solutions. This shows that the 3D neutron streaming effects that occur in the TREAT reactor are accurately estimated using PROTEUS-MOCEX. Such accurate estimation of PROTEUS indicates that we can move on its validation tests with the same experimental reactors in FY17.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Under the U.S. DOE NEAMS program, the high-fidelity multi-physics modeling and simulation capability SHARP [1] for nuclear reactor design and analysis has been developed. SHARP is a suite of physics simulation software modules and computational framework components that enables users to accurately evaluate the physical processes of nuclear reactors including neutron transport, thermal fluid, and fuel and structure behaviors. Among the SHARP components, the goal of the NEAMS neutronics effort is to develop a neutronics toolkit for use primarily on sodium-cooled fast reactors (SFRs) and an extended use to other reactor types. The neutronics toolkit includes the high-fidelity deterministic neutron transport code PROTEUS [2] and many supporting tools such as a cross section generation code MC^2-3 [3], a cross section library generation code Genesis, alternative cross section generation tools, mesh generation and conversion utilities, an automated regression test tool BuildBot, and a reactivity perturbation and sensitivity analysis code PERSENT [4], as shown in Figure 1.

The FY16 effort for NEAMS neutronics focused on supporting the release of the SHARP toolkit and existing and new users, continuing to develop PROTEUS functions necessary for performance improvement as well as the SHARP release, verifying PROTEUS against available existing benchmark problems, and developing new benchmark problems as needed. The efforts of updating PROTEUS will eventually support the SHARP mission to provide advanced tools for a variety of nuclear reactor cores, and promote its use by DOE programs and the nuclear industry.

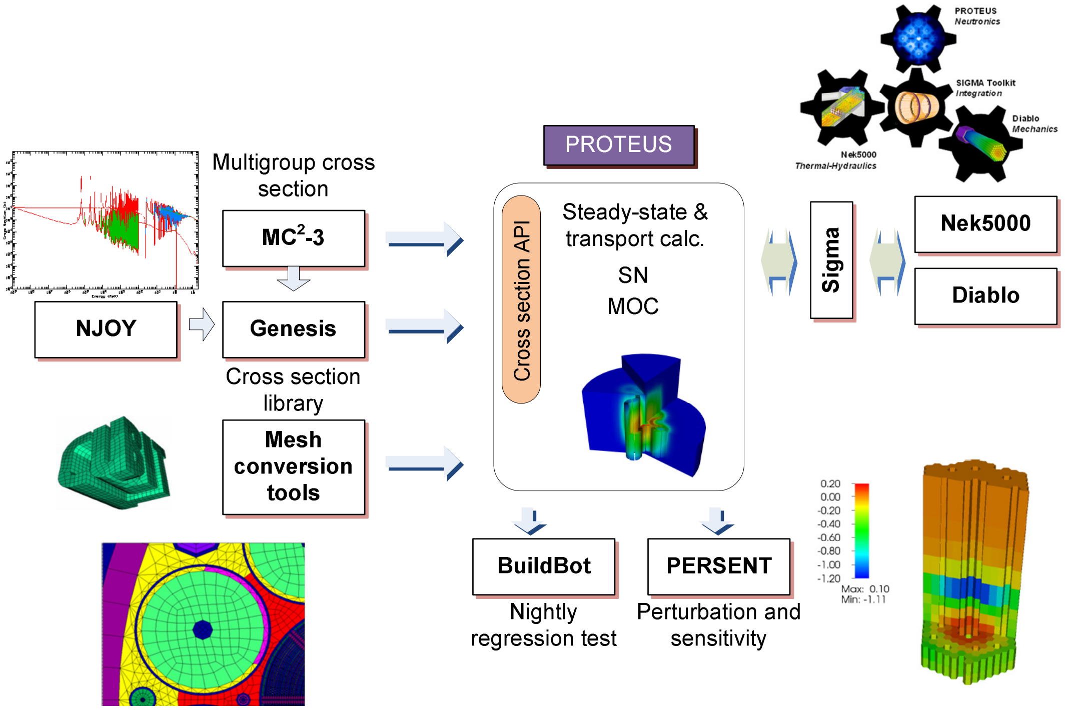


Figure 1. NEAMS Neutronics Tools in Support of SHARP.

In order to support the SHARP toolkit release, the PROTEUS-SN solver was updated to perform kinetics calculations driven by multi-physics coupled calculations within SHARP. Additionally, mesh deformation simulations were streamlined with automatic material density and concentration updates now being performed from inside PROTEUS-SN automatically based on mesh deformation and material models. This new automatic density capability supports multi-physics calculations involving geometry deformation. The new standalone version of PROTEUS-SN was released at the same time as the SHARP release in March 2016. Additional support and development on PROTEUS-SN has been performed based on requests by users.

Outside of SHARP, PROTEUS-SN can now automatically analyze deformed mesh simulations using some of the logic implemented for SHARP calculations. The new mesh deformation capability was verified against legacy procedures to compute reactivity feedback for a number of assumed deformed configurations for the ABTR design. Interestingly, the legacy procedures were shown to exaggerate the reactivity feedback due to radial core expansion by 35% compared to direct simulation. This provides motivation for direct 3D simulation using a high fidelity code like PROTEUS-SN.

Updates were made to the PROTEUS-MOCEX solver [5] which is focused on solving 3D extruded geometry problems using the 2D MOC method coupled with a Galerkin discontinuous finite element method axially. Improvements include changing the solution method across energy groups to reduce memory requirements and improve computational efficiency.

For more accurate cross section generation, the 2D MOC capability of MC^2-3 developed in FY15 is expanded to 3D using the MOCEX method of PROTEUS. A consistent CMFD formulation is applied to MOCEX for effective acceleration, which would be a preliminary testing of CMFD for its implementation to PROTEUS-MOCEX in future. Initial testing of the new CMFD formulation is performed for both thermal and fast reactor test problems with different problem sizes.

As an alternative cross section generation option, the approach using cross sections generated directly from the Serpent Monte Carlo code [6] is updated to use the OpenMC Monte Carlo code [7]. The main idea of the alternative cross section generation option is to provide users with various options for cross section generation, such as MC^2-3, online cross section generation using the cross section API, and Monte Carlo codes, so that they can select an affordable option fit to their problems of interest.

A further development of PROTEUS functions and capabilities is continued to accurately simulate various reactor types including sodium-cooled fast reactors and thermal reactors associated with ongoing NEUP projects. This includes the improvement of transient solver along with transport solvers as needed, numerical acceleration to improve computational efficiency, implementation of fast-running solvers for multi-resolution solutions, and improvement of user interfaces based on user requests.

For code verification and validation efforts, two thermal reactor problems, TREAT [8] and Reactor Critical Facility (RCF) at RPI [9], are modeled and simulated using PROTEUS. Both are challenging problems in terms of geometry complexity. Especially, the air cooling channels through chamfered areas and hodoscope causes the strong neutron streaming in the axial direction which makes significant impact on global and local solutions. The MOCEX solver is used to accurately deal with the 3D neutron behavior for such a reactor.

Section 2 presents improvements made to NEAMS neutronics codes including PROTEUS, MC^2-3, mesh utilities, an automated regression test tool BuildBot, PERSENT, and an alternative cross section generation tool. Section 3 discusses verification problems and results from PROTEUS modeling and simulation for SFR and thermal reactors. Section 4 presents user support activities carried out during FY16. Conclusions and future work are discussed in Section 5.

2. Improvements in NEAMS Neutronics Codes

During FY16, numerous improvements have been made to the neutronics codes developed under NEAMS. The PROTEUS-SN solver was updated to perform kinetics calculations as well as mesh deformation simulations as part of the SHARP multiphysics toolkit. The PROTEUS-MOCEX solver improvements include changing the solution method across energy groups to reduce memory requirements and improve computational efficiency, and development of a consistent CMFD formulation for effective acceleration. Improvements to MC^2-3 include development of 3D MOC capability based on the MOCEX method, extension of the cross section library to the thermal energy range, and update of gamma libraries with 94 energy groups. An alternative cross section approach using the Serpent Monte Carlo code is expanded to include the OpenMC Monte Carlo code. Mesh utilities and BuildBot as well as PERSENT have been improved to support NEAMS neutronics simulations.

2.1 PROTEUS-SN

The PROTEUS-SN solver discretizes the second order even parity form of the transport equation using discrete ordinates and continuous finite elements. PROTEUS-SN is the neutronics module with the SHARP multi-physics toolkit and is therefore coupled to other physics code. Significant development was performed on PROTEUS-SN to verify and enhance multiphysics coupling features in SHARP. First, the PROTEUS-SN code was transitioned out of the Nuclear Engineering code repository and into the SHARP repository. In the past, having two separate repositories caused many issues with code maintenance. Starting this year, the SHARP repository is the development location for PROTEUS. After moving the code, the PROTEUS-SN workflow was first verified for both standalone and coupled problems within SHARP. Significant debugging and verification was performed to ensure consistency. PROTEUS-SN was updated to use newer versions of external libraries (Metis 5.0+, MPICH 3.2+, and PETSc 3.4) for compatibility with SHARP. New physics features were then added in order to enable new types of calculations as well as avoid tedious user processing for certain types of problems. These updates are described below.

2.1.1 Integration of Kinetics Capabilities into SHARP

The PROTEUS-SN kinetics capability is now fully integrated into SHARP. Previously only steady state calculations were possible with SHARP. Connections were made to the kinetics routines such that time-dependent calculations can be made for simple transients (material changes such as control rod insertions are not permitted, for example). The user needs only to pass an input flag to the SHARP executable “-proteus_kinetics” as well as provide a simple kinetics input file in the local directory. In kinetics mode, the timesteps are fully controlled by the Nek5000 thermal hydraulics solver. The transients are driven by temperature and density feedback from the other physics modules. A time-dependent power curve generated using the PROTEUS-Nek and PROTEUS-Nek-Diablo drivers of SHARP with kinetics enabled is plotted in Figure 2. The two PROTEUS-Nek-Diablo calculations differ in the use of sodium backfill calculations – the blue line employs sodium backfill in expanded regions with sodium whereas the red dashed line expands sodium at the same rate as the expanded geometry.

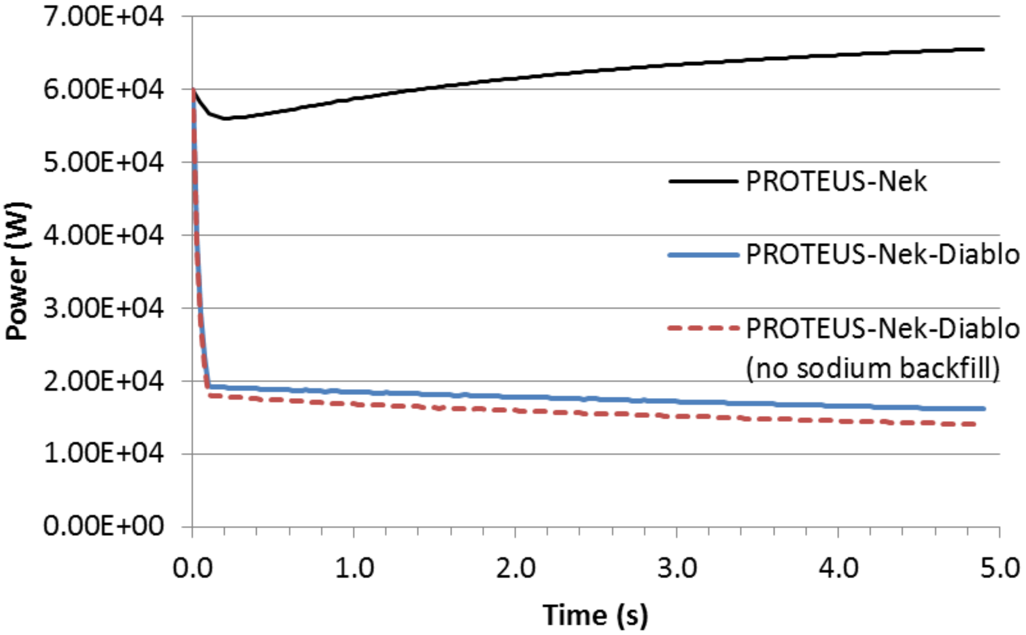


Figure 2. Power Evolution with Time for SAHEX (Single Hex Assembly) Problem with Different SHARP Multiphysics Drivers.

2.1.2 Automatic Density Updates for SHARP Simulations with Structural Feedback

In order to facilitate the user input procedure for multiphysics calculations involving structural deformations, a new capability was implemented in PROTEUS-SN to automatically process material densities to preserve mass. On a block-wise basis, PROTEUS-SN now automatically calculates the change in block volume due to deformation and adjusts the local isotopic densities to preserve mass. Furthermore, material models are now available to treat certain isotopes specially. For example, a sodium material model exists whereby sodium density is calculated based on temperature. This model allows the simulation of sodium backfill into any expanded/contracted regions rather than treating sodium like a thermally expanded solid material. Any new material model can easily be added in the future. This automatic density computation was extended later in the year to apply to PROTEUS-SN standalone simulations, and not just SHARP coupled simulations.

2.1.3 Documentation, Testing, and Output Improvements for SHARP

To improve the user output, enhanced information is now printed during multiphysics simulations. Block-wise temperatures, densities, and volumes are printed between physics iterations so that the user can track physics feedback as the calculation proceeds. Additionally a

SHARP manual was created which now includes a section on how to set up PROTEUS-SN neutronics problems for SHARP coupled calculations and a description of PROTEUS-SN input file requirements for SHARP calculations. These include highlighting special features of the code which can be enabled as well as requirements for multigroup cross section data file format (must exist at two temperatures). Finally, verification problems for PROTEUS-SN (both standalone and coupled) were added to the SHARP repository for user verification of correct installation. Work is ongoing to further enhance SHARP by including the Cross Section API (CSAPI) as a compilation option.

2.2 PROTEUS-MOCEX

The PROTEUS-MOCEX code solves the 3D transport equation using method of characteristics (MOC) in the 2D X-Y plane and discontinuous Galerkin finite elements in the Z dimension. During FY16, improvements were made to the performance and memory requirements of the PROTEUS-MOCEX solver by implementing a Gauss-Seidel solution scheme in energy and developing a consistent coarse mesh finite difference acceleration scheme that is compatible with the solver.

2.2.1 Group-Sweeping Solution Scheme

The original PROTEUS-MOCEX code solved the discretized multi-group transport equation using the GMRES method, which is one of the Krylov subspace methods. To explain, we assemble the space-angle-energy terms into the flux vector Ψ and source Q (fission and/or fixed) such that we can write:

$$\begin{aligned}&(\mathbf{L}-\mathbf{W})\Psi = Q,\\ &(\mathbf{I}-\mathbf{L}^{-1}\mathbf{W})\Psi = \mathbf{L}^{-1}Q, \ .\\ &\mathbf{A}\Psi = b.\end{aligned} \tag{1}$$

In this system, the loss and collision operator $\mathbf{L}$ is inverted as part of the $\mathbf{A}$ matrix definition which is done because it is block diagonal with respect to energy and minimizes the vector size of Ψ . This leaves the remaining inversion of the scattering operator, $\mathbf{W}$, to be iteratively converged. The original intention of the MOCEX implementation was to develop an appropriate preconditioner using conventional Gauss-Seidel iteration in energy where a simplified within group diffusion representation of the scattering operator would be constructed. The above approach is advantageous for parallel computation since energy group decomposition can be made scalable simultaneously with space and angle decomposition.

With an un-preconditioned GMRES method, the number of back vectors required to solve the system can become quite high. The set of back vectors are defined using a Gram-Schmidt orthogonalization procedure and require a certain span (number of them) in order to prevent oscillation about local minimum or divergence. In the implementation, the GMRES method is applied to a multi-group equation and the basis vectors span the entire space, angle and energy domains which makes each back vector considerably large. When solving neutronics problems with PROTEUS-MOCEX, we have observed the need to use between 30 and 100 back vectors where we had hoped the preconditioned GMRES approach would only require 5-10. Without preconditioning, excessive memory is required to accommodate the large basis vector sets, which strongly limits the applicability of the code to real world transport calculations.

In PROTEUS-MOCEX, energy decomposition is not currently scalable and requires additional research to understand the fundamental implementation problem. In a classical iterative approach in neutronics, the multigroup system is solved using a Gauss-Seidel scheme in energy. In that approach, the error norm is typically focused on the eigenvalue and fission source convergence because of using the power method. In that situation, the individual energy group equations contribute to the error norms in disproportional amounts. As an example, in fast spectrum systems, the lowest energy groups typically have no impact on the fission source or eigenvalue and can be loosely converged relative to the energy groups that make up the bulk of the neutrons in the system. Further, the condition number of each diagonal system, i.e. the within group transport equation, is dependent upon the scattering ratio and optical properties (total cross section) of the energy group. Additionally, because of parallel decomposition, the condition number of each within group equation is modified by the optical thickness of each processors assigned subdomain and the total number of subdomains. In this situation, the amount of computational work required to solve each within group system can be considerably different from that observed in serial (i.e. because of the parallelism) and because of the physics properties.

In a conventional, non-parallel implementation, a relative error criteria applied to the within group equation in the Gauss-Seidel scheme that is weighted with respect to the fission source contribution can be shown to lead to the minimal computational effort in an un-accelerated power method. In the un-preconditioned GMRES solver of PROTEUS-MOCEX, there is no similar physical control that can be applied to the space of $\mathbf{A}\Psi$ that is consistent with the physically interpretable space of Ψ . Thus without a preconditioner that can account for the relative importance of the components of Ψ , an un-preconditioned GMRES approach is woefully inefficient compared with a Gauss-Seidel approach to solving the system of equations with or without parallelism.

Because of the time constraints on the development of PROTEUS-MOCEX, we have implemented the conventional Gauss-Seidel iterative approach in energy for the time being. This simultaneously reduces the memory requirements as it eliminates the GMRES solver applied to the entire space and improves the performance as Gauss-Seidel is almost always the optimal scheme for the multigroup system when no parallelism is being employed. PROTEUS-MOCEX was already equipped with a within-group solver as it is required for the fixed source problems in the sub-group method. In this case, the within group equation has a form identical to Eq. (1) except for the understanding that the $\mathbf{L}$ matrix is the group-wise loss and collision operator, $\mathbf{W}$ is the within-group scattering operator, Q is the group source, and Ψ is the mesh-averaged angular fluxes for a given energy group. In this implementation, the GMRES solver is applied to the group-wise form of Eq. (1).

In the parallel computation with domain decomposition, Eq. (2) is modified to account for the communication between the neighboring domains explicitly in the formulation as:

$$(\mathbf{I} - \mathbf{L}_i^{-1}\mathbf{W}_i\mathbf{K}_i)\Gamma_i = \mathbf{L}_i^{-1}Q_i \quad , \tag{2}$$

where i is the index for the processor, Γ_i is the new solution in the parallel computation that additionally includes the boundary angular flux information, and $\mathbf{K}_i$ is the communication operator.

In order to employ the Gauss-Seidel scheme using the existing within-group solver, the group source construction including fission and scattering reactions was added to the PROTEUS-MOCEX code. Note that the within-group scattering is excluded in the group source computation since it is internally updated in the within-group solver as shown in Eq. (2). In the conventional calculation procedure of PROTEUS-MOCEX, the Gauss-Seidel scheme was added yielding three layers of nested iterations: the fission source iteration (power method), Gauss-Seidel iteration over energy, and the within-group Krylov subspace iteration (each diagonal inversion of Gauss-Seidel).

The updated version of PROTEUS-MOCEX code is equipped with a multi-group Krylov (MGK) solver and a Gauss-Seidel (GS-WGK) in energy scheme with a within-group Krylov (WGK) solver. In the PROTEUS-MOCEX calculations, the default flux solver is the MGK solver. Alternatively, the GS-WGK solver can be invoked in the flux calculation when this solver is specified in an input file. In order to test the newly implemented group iteration scheme in the PROTEUS-MOC code, performance and accuracy was analyzed for three test problems based on the C5G7 benchmark [10]: 2D assembly, 2D core and 3D core problems. The 3D core problem was taken from the 3D core configuration specified in the C5G7 benchmark. The 2D core problem was obtained by taking a slice of the 3D core problem along the radial direction at the bottom of the core. The 2D assembly problem was derived from the center assembly of 2D problem, which is filled with UO_2 fuels. The 7-group cross sections provided in the specification of the C5G7 benchmark were assigned to individual regions of three problems.

In order to examine the impact of the GS scheme on memory usage and computational efficiency, the PROTEUS-MOCEX calculations were performed using the MGK and GS-WGK solvers. We note that the GS-WGK approach for thermal spectrum systems typically utilizes “up-scattering” iterations over the thermal energy range to improve the convergence rate of GS. This is a physics based decision where we know that the higher energy groups mostly contribute neutrons to the thermal range (rather than the fission source) and that most of the neutrons of importance are in the thermal range. Table 1 summarizes the results for the three test problems. The eigenvalue obtained with the GS-WGK solver are identical to the reference MGK solver results, which confirms that the GS scheme was properly implemented in the existing framework of PROTEUS-MOCEX code. As shown in Table 1, the memory usage for the Krylov solver is significantly reduced by about a factor of six because the array size for storing the back vectors, which dominates the memory usage, is reduced by a factor of the number of groups. Since the

GS-WGK solver reduces the computational time for solving multi-group problem with the given fission source, the total computational efficiency is notably improved as high as a factor of 2.4 for the 3D core problem. The number of fission source iterations increases slightly due to the upscattering.

Table 1. Summary of Performance Examinations

Problem		2D Assembly		2D Core		3D Core	
Solver		MGK	WGK	MGK	WGK	MGK	WGK
Domain Decomposition	Angle	16		16		16	
	Radial	5		20		4	
	Axial	-		-		5	
# of Processors		80	80	320	320	320	320
Eigenvalue		1.33429	1.33429	1.18646	1.18646	1.14558	1.14558
# of Fission Source Iterations		10	11	41	43	42	45
Total Computing Time, Sec		12.0	8.9	712	391	6188	2569
# of Back Vectors		20		20		20	
Memory Usage for Krylov Solver[1)]		35.8 Mb	6.7 Mb	295 Mb	49.1 Mb	2.8 Gb	458 Mb
Computing Time Reduction		1.35		1.82		2.40	
Memory Reduction for Krylov Solver		5.34		6.01		6.26	

2.2.2 *Preliminary Implementation of CMFD to MOCEX*

2.2.2.1 CMFD Formulation for MOCEX

The Coarse Mesh Finite Difference (CMFD) scheme satisfies the neutron balance equation in conjunction with the corrective current relation that preserves the leakage of the original higher order solution. Therefore, in order to use the CMFD acceleration scheme, the spatial discretization employed in the transport solver should strictly satisfy the global and local neutron balances. Because of the non-conservative discretization property of the Galerkin method applied to the axial variable in the MOCEX method, a non-conservative discretization results and the CMFD prerequisite is not satisfied. A blind application of the CMFD scheme results in inconsistency of the CMFD solution with the MOCEX solution. To enforce neutron balance in each coarse mesh and consistency between CMFD and higher order equations, a consistent CMFD formulation for MOCEX calculation was devised by introducing a fictitious cross section, referred to as a pseudo absorption cross section, in each CMFD neutron balance equation.

In order to formulate a CMFD problem for accelerating the 3D transport calculation, the following quantities are needed: the homogenized group constants for coarse meshes and the coupling coefficients that specify interface current relations between the two adjacent coarse meshes. From the transport solution, homogenized group constants can be generated by flux-volume weighting over the fine meshes belonging to the coarse mesh. The coupling coefficients can be determined using the following CMFD relation:

$$J_{i\to k} = -\tilde{D}_{i\to k}(\bar{\phi}_k - \bar{\phi}_i) - \hat{D}_{i\to k}(\bar{\phi}_k + \bar{\phi}_i) \ , \tag{3}$$

where i and j are indices for the coarse meshes that are connected through a common interface. $\bar{\phi}_i$ and $\bar{\phi}_j$ is the averaged fluxes of the coarse meshes i and j, respectively, and $J_{i\to k}$ is the surface current from the coarse mesh i to the coarse mesh j. The average flux and surface current can be readily obtained from the MOCEX transport solution. $\tilde{D}$ is the coupling coefficient obtained from the conventional finite difference diffusion formulation, and $\hat{D}$ are the current correction factor to preserve the interface current obtained from the MOC solution. The CMFD relation in Eq. (3) can be solved for the current correction factor as:

$$\hat{D}_{i\to k} = -\frac{J_{i\to k} + \tilde{D}_{i\to k}(\bar{\phi}_k - \bar{\phi}_i)}{(\bar{\phi}_k + \bar{\phi}_i)} \ . \tag{4}$$

Using the surface currents obtained from the MOCEX solution in Eq. (4), the current correction factors can be determined. Under the current continuity condition, the current correction factors satisfy the following relation:

$$\hat{D}_{i\to k} = -\hat{D}_{k\to i} \ . \tag{5}$$

As a remedy to enforce the neutron balance and the consistency of the CMFD equation, a fictitious cross section, referred as a pseudo absorption cross section, is introduced in the CMFD neutron balance equation as:

$$\sum_k J_{i\to k} + (\Sigma_r^i + \Sigma_p^i)\bar{\phi}_i V_i = \bar{q}_i V_i \ , \tag{6}$$

where Σ_p is the pseudo absorption cross section. The pseudo absorption cross section of each coarse mesh can be determined from the MOCEX solution as:

$$\Sigma_p^i = \frac{\bar{q}_i V_i - \sum_k J_{i\to k}}{\bar{\phi}_i V_i} - \Sigma_r^i \ . \tag{7}$$

The homogenized group constants, current correction factors and pseudo absorption cross sections are iteratively updated during the iterative solution process because those parameters are a function of the transport solution.

In the MOCEX method, the angular flux and current can be discontinuous at the mesh interfaces. In order to secure the consistent CMFD current relation, the discontinuity of current is inevitably allowed and consequently, Eq. (5) does not hold for the current correction factors. Therefore, when the correction factor for the current from a coarse mesh i to a coarse mesh k, $\hat{D}_{i\to k}$, is determined, the current value obtained from the angular flux distribution in the coarse mesh i should be used. Alternatively, a single current at the coarse mesh interface can be defined by making use of the outgoing angular fluxes of two neighboring coarse meshes at the common interface [11] and this current is used in the determination of the current correction factor in Eq. (4) and the PAXS in Eq. (7).

2.2.2.2 Modification of the MOCEX Solver of MC²-3

A development test code based upon the methodology used in PROTEUS-MOCEX, but focused on a classical semi-structured geometry approach, common for MOC codes, was created in the past year. The primary test focus of that code was with MC^2-3 to perform hexagonal unit cell calculations in a triangular lattice. It was later extended to handle typical SFR core configurations. This 3D transport solver of MC^2-3 could be utilized for testing the newly developed CMFD acceleration method for the MOCEX solver except that the size of the ultra-fine group structure of MC^2-3 does not allow us to use it on large problems. Therefore, the MOCEX transport solver of MC^2-3 was converted into a standalone transport code such that it could perform transport calculations using pre-generated macroscopic cross section sets with a coarser set of energy groups (i.e. similar to the TWODANT procedure of MC^2-3).

The CMFD acceleration is a diffusion level calculation applied to the coarse mesh geometry. In a typical SFR core configuration, hexagonal assemblies or hexagonal pin cells would be chosen as the coarse meshes. A module for defining the coarse mesh geometry was created to provide the mesh connectivity needed in the construction of linear systems for CMFD acceleration. Along with the coarse mesh geometry, the neutron current information at the coarse mesh interfaces is required in the determination of the CMFD coupling coefficients. A radial surface current at the assembly or pin interfaces can be computed during the MOC calculation by using the following equation:

$$J_{i\to k}=\sum_{l\in s}\sum_{m}\varphi^{l,out}_{m,\mathrm{i}}\sin\theta_m w_m\Delta_m-\sum_{l\in s}\sum_{m}\varphi^{l,in}_{m,k}\sin\theta_m w_m\Delta_m \;, \tag{8}$$

where m, l, and s are the indices for the angle, tracking ray, and common surface of coarse meshes i and k, respectively. φ is the angular flux, w is the weight of angular quadrature, and Δ is the ray spacing. An axial surface current can easily be computed using the mesh-averaged angular flux solution, obtained as part of the MOC calculation for each plane.

2.2.2.3 Overall Calculation Flow

The overall calculation procedure is illustrated in Figure 3. In the MOCEX calculation with CMFD acceleration, the CMFD system is formed with the partially converged transport solution from a single Gauss-Seidel (in energy) iteration. For strong up-scatter problems, the up-scattering block may require additional iteration (i.e. effectively two or three GS iterations). The CMFD problem is solved using a conventional matrix free iterative scheme (GS in energy, within group GS in space). From Figure 3, the CMFD acceleration is cast as its own eigenvalue problem and is solved using the power method. The fine mesh MOCEX flux solution is updated by applying the ratio of the coarse mesh flux from the previous partially converged transport solution and the newly obtained CMFD solution. The updated fine-mesh solution (and eigenvalue) is then used in the next GS iteration on the fine mesh solution. It is important to note that the power method is not applied to the fine mesh system, only the coarse mesh system.

With regard to stability, the within-group iteration on the fine mesh MOCEX system must be done sufficiently to guarantee that the CMFD system is physical (i.e. converges). The current implementation uses a fixed iteration algorithm where the user input specifies the fixed number of iterations to be applied within each Gauss-Seidel iteration. We are researching a dynamic convergence approach for future implementations to further reduce the computational effort.

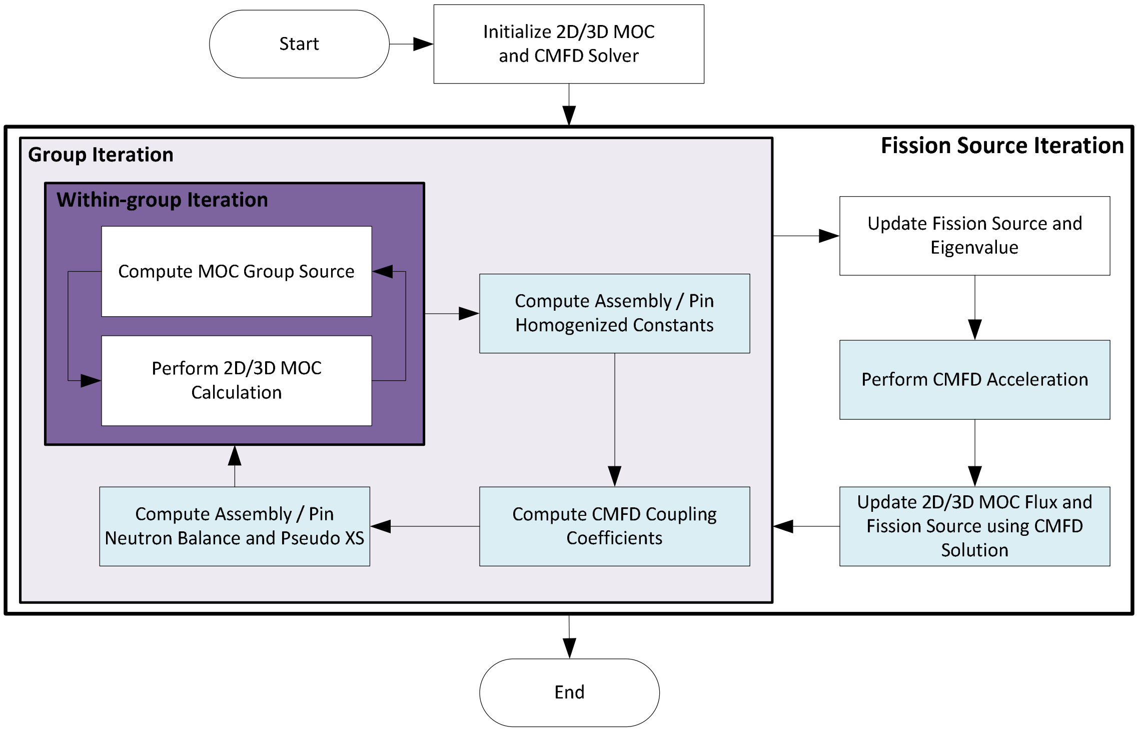

Start
Initialize 2D/3D MOC and CMFD Solver
Group Iteration
Fission Source Iteration
Within-group Iteration
Compute MOC Group Source
Perform 2D/3D MOC Calculation
Compute Assembly / Pin Homogenized Constants
Update Fission Source and Eigenvalue
Perform CMFD Acceleration
Compute Assembly / Pin Neutron Balance and Pseudo XS
Compute CMFD Coupling Coefficients
Update 2D/3D MOC Flux and Fission Source using CMFD Solution
End

Figure 3. Calculation Flow of MOCEX Solver with CMFD Acceleration.

2.2.2.4 Performance of CMFD Acceleration with MOCEX

Two test problems were prepared to test the application of CMFD acceleration to the MOCEX calculation. Hexagonal core configurations were selected due to the limited geometry handling capability of the test code. Cross section data for thermal and fast reactor types were applied while the geometric configurations are identical.

The assembly and core configurations of those problems are given in Figure 4 and Figure 5. The assembly pitch is 6.77 cm, and each assembly contains 19 fuel pins with 1.08 cm diameter. Axially, each assembly consists of one 28.56 cm fuel region and one 21.42 cm reflector region with the reflective and vacuum boundary conditions at the outer ends of the fuel and reflector regions, respectively. The model consists of 10 axial meshes in total. The pin radius and axial sizes were adopted from the C5G7 benchmark specification [10]. The hexagonal assemblies in Figure 5 were arranged in a 6-ring triangular lattice, and one-third core problems were solved by imposing radial periodic boundary conditions.

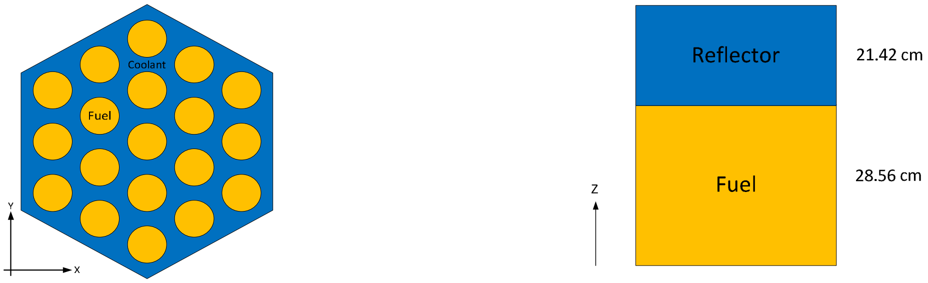
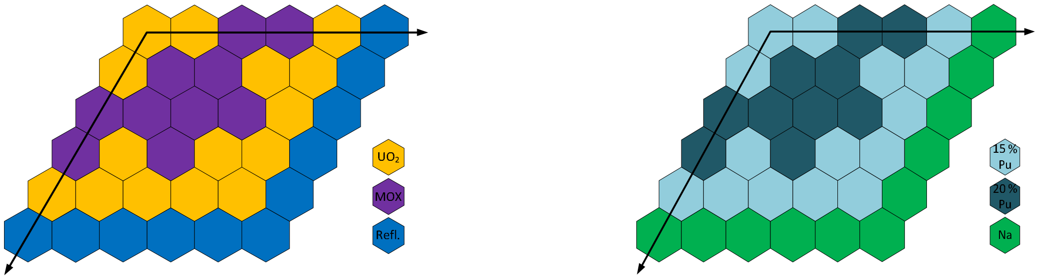


Figure 4. Assembly Configuration of Test Problems.



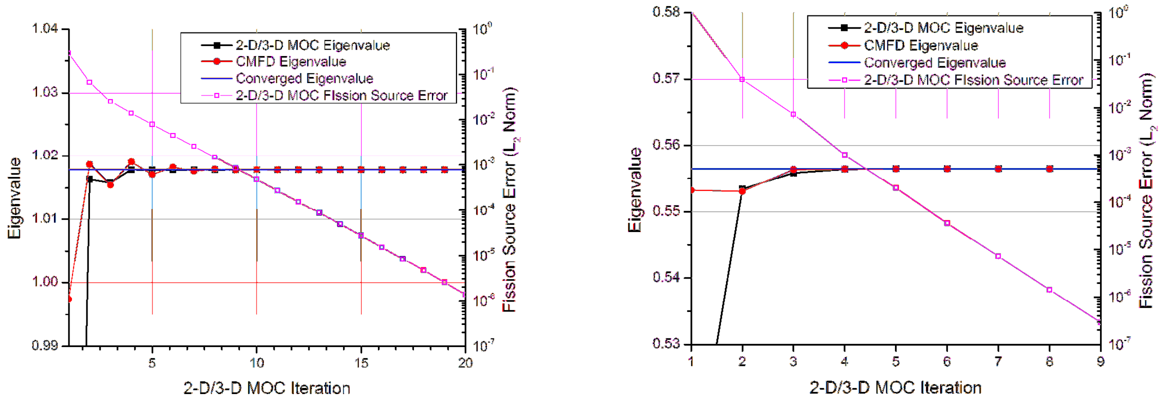
(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 5. Core Configuration of Test Problems.

The 7-group cross section set for the thermal reactor problem was taken from the C5G7 benchmark problem. For the fast reactor problem, the 9-group cross section data based on the ABTR cell model was generated using the MC^2-3 code. In the MC^2-3 run, the edit option for region-wise cross section generation was used and the transport correction was applied to the P_0 broad-group scattering matrix. Reference solutions were not obtained using an independent code such as MCNP-6 because the purpose of this work is not to verify accuracy of transport solutions but to demonstrate the feasibility the CMFD acceleration. Thus the solutions obtained with CMFD acceleration are compared directly to the solutions without acceleration.

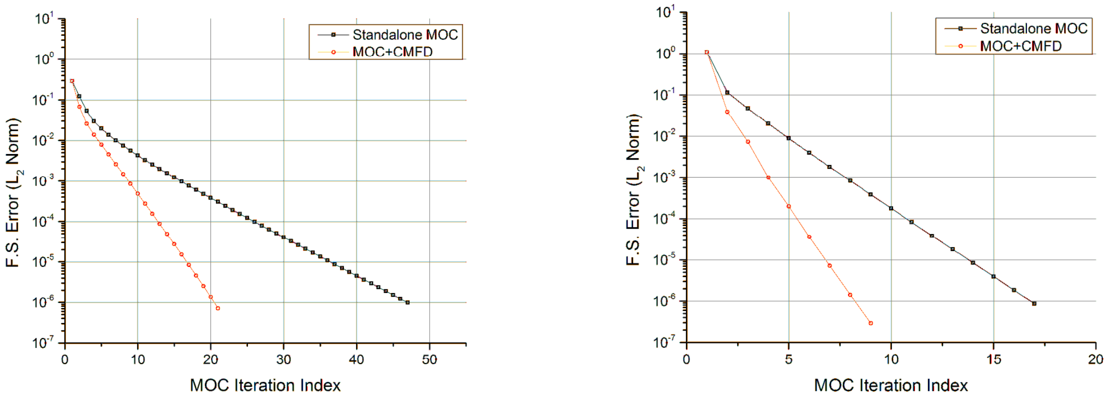
In the tests performed here, each assembly was assigned to a single coarse mesh, which is referred as the assembly-based CMFD calculation, and a tightened convergence criterion for MOCEX solutions, 10^{-6} of fission source error (L_2 norm), was imposed. The convergence behaviors of eigenvalue and fission source for the thermal and fast reactor test problems are given in Figure 6 and Figure 7. The eigenvalues of both cases converge rapidly within 1 pcm difference after 5 fission source iterations on the MOCEX calculation, and the fission source errors decrease monotonically in both cases. The eigenvalue error between the accelerated and un-accelerated solution decreases with each iteration and those two eigenvalues are observed to converge to the same solution. Figure 7 compares the fission source errors for standalone and CMFD coupled runs. The number of fission source iterations, which determines the total computing time, is reduced by a factor of two for both the thermal and fast reactor problems.



(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 6. Convergence Behavior of MOCEX Calculation with CMFD Acceleration.

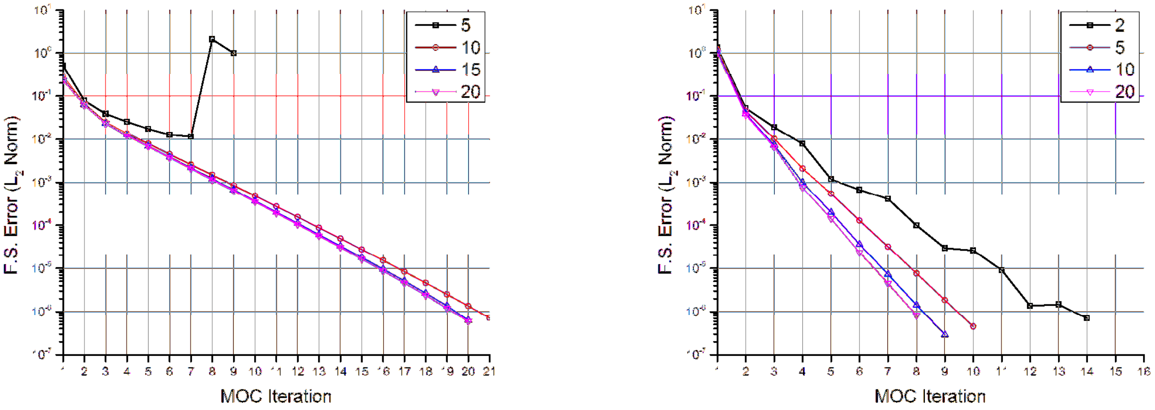


(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 7. Comparison of Fission Source Convergence.

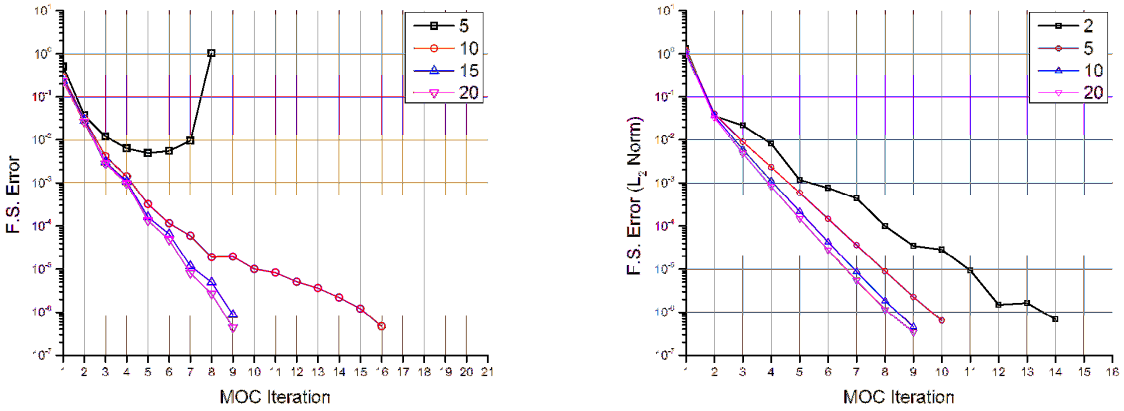
Calculations using assembly-based and pin-based CMFD acceleration were performed with different numbers of within-group iterations to investigate the impact on convergence rates. The results are summarized in Figure 8 and Figure 9. The thermal reactor problem with pin-based CMFD converges significantly faster than the assembly-based CMFD problem. However, the fission source convergence behavior becomes unstable when fewer than 10 WG iterations are performed. In summary, the pin-based CMFD calculation requires a larger number of WG iterations than the assembly-based CMFD calculation to obtain sufficiently accurate current correction factors. For the pin-based CMFD acceleration case using 15 WG iterations per single group iteration, the number of fission source iterations is reduced from 40 (standalone MOCEX method) to 9, a factor of ~4 improvement, while the assembly-based CMFD acceleration needs 20 iterations (a factor of 2 improvement).



(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 8. Fission Source Convergence of Assembly-based CMFD Acceleration with Different Number of WG Iteration.



(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 9. Fission Source Convergence of Pin-based CMFD Acceleration with Different Number of WG Iteration.

Since the pin-based CMFD acceleration scheme can improve the convergence of intra-assembly flux distribution, the pin-based CMFD calculation is superior to the assembly-based one for thermal reactor problems which have substantial flux variation within the assembly. For the fast reactor problem, the convergence behavior is similar for both the pin- and assembly- based coarse meshes because the flux variation within an assembly is not pronounced. These results suggest that optimal performance could be obtained when the CMFD mesh size is comparable to the neutron mean free path of the problem of interest. A method to determine an optimal number of within-group iteration would be required since the code would be used to solve large problems with parallel computation which usually degrades the convergence characteristics.

It has been observed that CMFD acceleration of neutron transport solutions is more effective for large core problems with a dominance ratio close to unity. The dominance ratio of the 3D test problems considered is relatively small compared to realistic core problems. Therefore, in order to demonstrate the effectiveness of the CMFD acceleration scheme, four additional 3D core configurations were prepared with different number of assemblies: 5, 9, 13 and 17 rings of assemblies as shown in Figure 4. The assembly configurations given in Figure 10 were also used for constructing these problems.

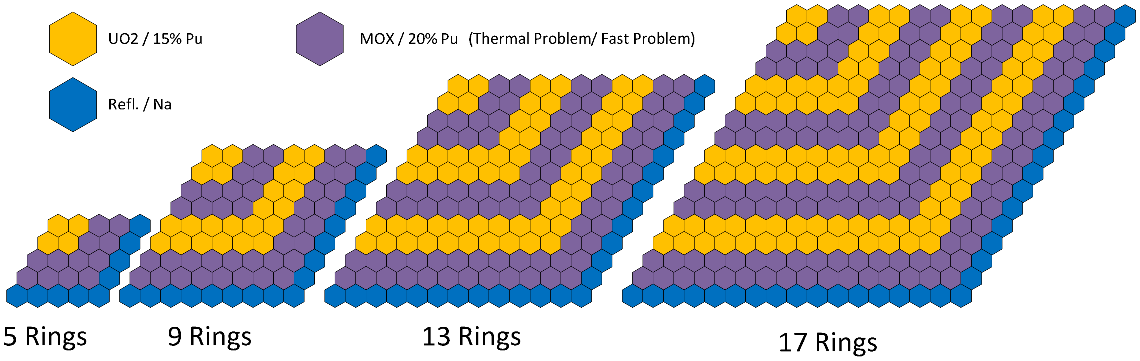
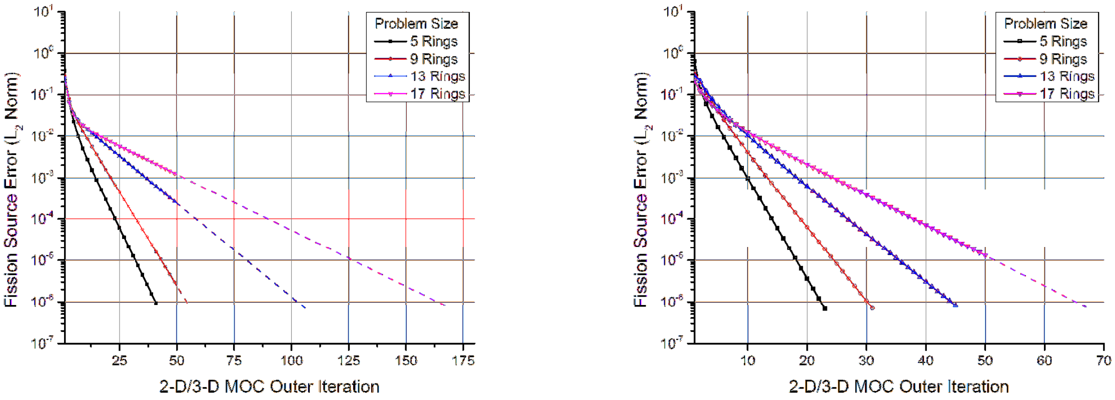


Figure 10. Core Configurations for Sensitivity Study on Problem Size.

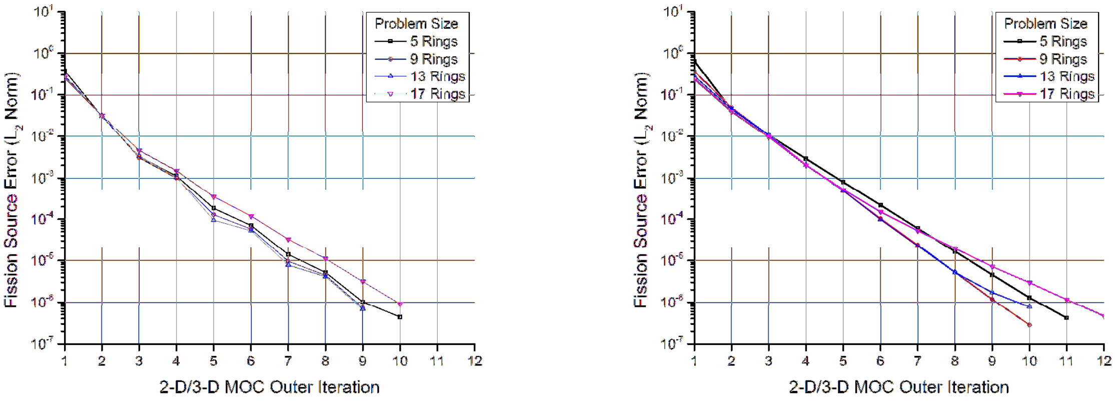
By solving these problems using the MOCEX solver with and without the CMFD acceleration scheme, the performance of the CMFD acceleration was systematically examined as a function of problem size. Pin-based CMFD with 15 and 5 fixed WG iterations per GS iteration were used for the thermal and fast reactor problems, respectively. A convergence criterion of 10^{-6} was used for the fission source iteration. For the reference calculations with no CMFD acceleration, the maximum number of fission source iterations was set at 50 to make the calculations practical. The error reduction behavior of the fission source beyond the maximum number of iterations and the number of fission source iterations required to satisfy the imposed criterion were predicted using the estimated dominance ratio. Figure 12 and Figure 11 show the convergence of the fission source for the cases with and without CMFD acceleration, respectively. The convergence behavior of without CMFD acceleration is significantly deteriorated and the number of fission source iterations grows rapidly as shown in Figure 11. On the other hand, as shown in Figure 12, the convergence with CMFD acceleration does not show any significant dependency on the problem size . For both thermal and fast reactor problems, the number of fission source iterations is around 10 regardless of the problem size. For the thermal reactor problem with the largest core size, the CMFD acceleration can reduce the computation time more than a factor of 15. These results show that the consistent CMFD formulation with pseudo-absorption cross sections (PAXS) is an effective acceleration scheme for the MOCEX calculations for realistic core problems.



(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 11. Fission Source Convergence vs. Problem Size Without CMFD Acceleration.



(a) Thermal Reactor Problem

(b) Fast Reactor Problem

Figure 12. Fission Source Convergence vs. Problem size with CMFD acceleration.

From the preceding, we can state that a consistent CMFD formulation usable with PROTEUS-MOCEX has been developed and verified to work on a test code. The acceleration technique has proven reliable and effective for test problems of varying size. More work is needed to implement the CMFD acceleration in the PROTEUS-MOCEX code because of the finite element based ray tracing and lack of a geometry hierarchy.

2.2.3 Improved Ray Tracing Scheme

For quite some time the PROTEUS-MOCEX code was observed to have significant problems producing consistent results when using spatial domain decomposition. After substantial debugging, a problem was found which allowed duplicated in parallel rays to exist thereby altering the physics of the problem being solved.

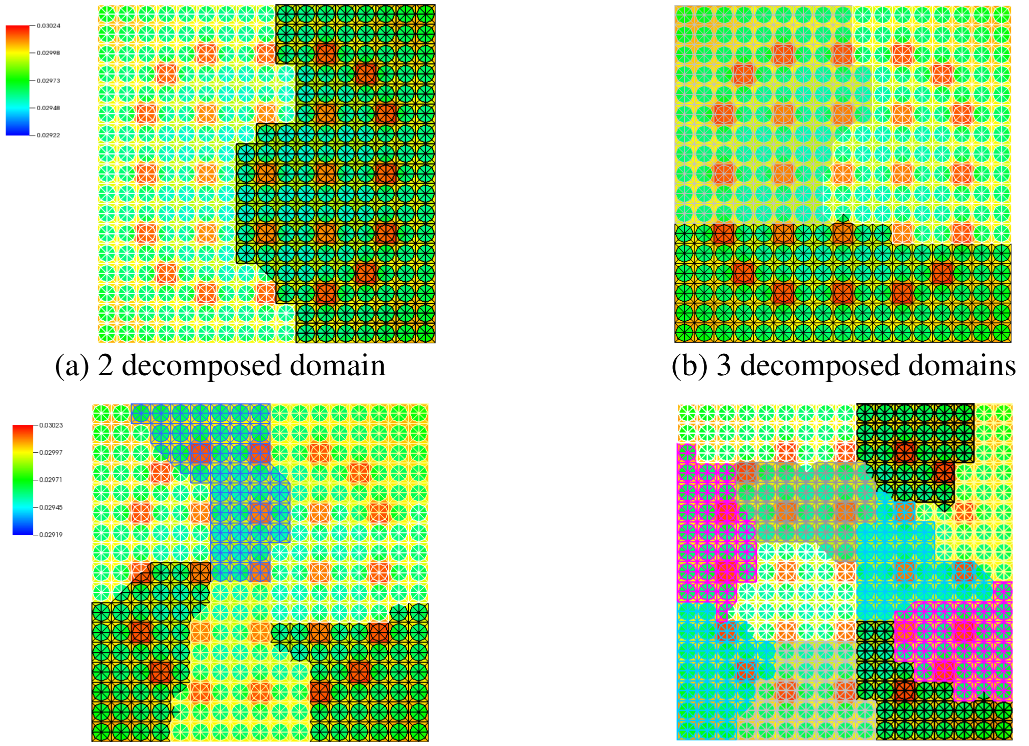
The simplest way to remove the ray duplication problem is to search for duplicated rays in the ray tracing data and remove them prior to assembling each global trajectory. After some study we decided that this approach would involve a considerable modification of the source code. As an alternative we realized that we could eliminate the problem by removing the starting points of duplicated rays prior to doing the local ray tracing. This approach requires very little work compared to the original scheme.

The starting point information of each trajectory does not provide sufficient information to determine if there will be a duplicated ray. Instead, if a trajectory has the same exit point from the domain but different starting points, there must be an overlapping section of trajectory data on that process. Therefore, the coordinates of the exit points provide sufficient information to detect trajectory duplication. When duplicated trajectories with the same exit point are detected, the ray to be excluded can be simply selected by comparing the location of the starting points. The algorithm to compute the trajectory exit points is adopted from the existing PROTEUS-MOC routine that generates the detailed ray tracing data and poses near negligible computational costs compared with the overall ray tracing time.

In order to verify the corrected PROTEUS-MOCEX code, a 2D heterogeneous assembly problem was re-calculated using the spatial parallel decomposition selections in Table 2. From Table 2, one can see that the updated PROTEUS-MOCEX code produces the correct eigenvalue for all calculations where we had observed considerable inconsistencies. Figure 13 shows that no discrepancies in the flux distribution are observed, regardless of the number of decomposed domains.

Table 2. Eigenvalues of 2D Assembly Problem Obtained Using Corrected PROTUES-MOCEX for Different Radial Domain Decompositions

Number of Decomposed Radial Domains	k-inf	Number of Decomposed Radial Domains	k-inf
1 (reference)	1.33429	9	1.33429
2	1.33429	10	1.33429
3	1.33429	11	1.33429
4	1.33429	12	1.33429
5	1.33429	13	1.33429
6	1.33429	14	1.33429
7	1.33429	15	1.33429
8	1.33429	16	1.33429



(a) 2 decomposed domain

(b) 3 decomposed domains

(c) 7 decomposed domains

(d) 11 decomposed domains

Figure 13. Scalar Flux Distribution of 2D Assembly Problem Obtained Using Corrected PROTEUS-MOCEX Code for Different Radial Domain Decompositions.

2.3 MC^2-3

In the previous year, the 2D transport capability of the MC^2-3 code was developed to represent the local heterogeneity effects more accurately for the unit cell of plate-type critical assemblies such as ZPR-3, ZPR-6 and ZPPR. In this year, work continued to extend the transport capabilities of MC^2-3. In the existing 2D MOC solver, the geometry modeling capability for hexagonal geometries was implemented such that the MC^2-3 code can perform 2D transport calculations for typical SFR assemblies. Along with this, the MOC solver was extended to incorporate a P_3 scattering kernel.

In the BFS facilities [12], a critical assembly is composed of stainless steel cylindrical tubes (wrapper tubes) vertically arranged in a triangular lattice, which are filled with cylindrical disks of different materials in a repeated cell arrangement. Often steel wires are inserted in the interstitial space among wrapper tubes. Due to the strong axial streaming along the interstitial space between the wrapper tubes, a 3D unit cell calculation is necessary to represent the local heterogeneity effect accurately. Moreover, an entire wrapper tube needs to be modeled explicitly in the 3-D transport calculation to reflect the axial leakage at the top and bottom of the repeated cells and the spectral transition at the interfaces of the axial reflector regions. The MOCEX test code described in the previous section was used because it can handle the 3D geometry of the BFS experiment and typical SFR assemblies.

2.3.1 Implementation of the MOCEX Solver

An efficient and robust 3D transport solver is desired to solve the slowing-down problems in the ultrafine group (UFG) or hyperfine group (HFG) structures. Currently, there are two deterministic methods applicable to 3D lattice problems with heterogeneous geometries. One method is the 2D/1D synthesis method originally developed for applications for pressurized water reactors. This method suffers from numerical instability problems when substantial axial leakage is present, and hence it was deemed not applicable to the BFS fast reactor problems. A full 3D MOC methodology requires excessive memory and computational resources and it too is impractical for use in ultrafine or hyperfine group transport calculations. The MOCEX methodology was therefore chosen as it provides the best alternative to the 2D/1D methodology that is known to have problems on this application area.

The derivation of MOCEX formulation is only sketched here and the detailed description can be found elsewhere [5]. The one-group neutron transport equation for a discretized angle is expressed as:

$$\left(\mu_m \frac{\partial}{\partial x} + \eta_m \frac{\partial}{\partial y} + \xi_m \frac{\partial}{\partial z}\right)\varphi_i^m(x,y,z) + \Sigma_i \varphi_i^m(x,y,z) = Q_i^m(x,y,z) \ , \tag{9}$$

where m and i are indices for angle and mesh, respectively. The axial variation of angular flux and source can be approximately represented with a finite set of orthogonal basis functions:

$$\varphi_i^m(x,y,z) \approx \sum_{j=1}^{N} \varphi_{i,j}^m(x,y) u_i^j(z) \ , \tag{10}$$

$$Q_i^m(x,y,z) \approx \sum_{j=1}^{N} Q_{i,j}^m u_i^j(z) \ , \tag{11}$$

where $u_i^j(z)$ is the j-th orthogonal basis function in the i-th mesh. By inserting Eqs. (10) and (11) into Eq. (9) and by applying the discontinuous Galerkin weighted residual technique, the following system of N equations are obtained for a polar direction in the upper hemisphere of angular domain ($\xi_m > 0$):

$$\begin{aligned}
&\left(\mu_m \frac{\partial \varphi_{i,k}^m(x,y)}{\partial x} + \eta_m \frac{\partial \varphi_{i,k}^m(x,y)}{\partial y}\right) < u_i^k(z), u_i^k(z) > + \xi_m u_i^k(z_i^+) \sum_{j=1}^{N} \varphi_{i,j}^m(x,y) u_i^j(z_i^+) \\
&\quad - \xi_m \sum_{j=1}^{N} \varphi_{i,j}^m(x,y) < u_i^j(z), \frac{\partial u_i^k(z)}{\partial z} > + \Sigma_i \varphi_{i,k}^m(x,y) < u_i^k(z), u_i^k(z) > \\
&\quad = Q_{i,k}^m < u_i^k(z), u_i^k(z) > + \xi_m \varphi_i^{m,-}(x,y) u_i^k(z_i^-) \quad (k = 1, \cdots, \mathrm{N}),
\end{aligned} \tag{12}$$

where $<\cdot,\cdot>$ denotes the integration of axial variable over the height of the mesh and $\varphi_{i,k}^{m,-}(x,y)$ is the incoming angular flux at the bottom of the i-th mesh. For the opposite angular domain ($\xi_m < 0$), a set of equations can be derived as:

$$\begin{aligned}&\left(\mu_m \frac{\partial \varphi_{i,k}^m(x,y)}{\partial x} + \eta_m \frac{\partial \varphi_{i,k}^m(x,y)}{\partial y}\right) < u_i^k(z), u_i^k(z) > -\xi_m u_i^k(z_i^-) \sum_{j=1}^{N} \varphi_{i,j}^m(x,y) u_i^j(z_i^-) \\ &-\xi_m \sum_{j=1}^{N} \varphi_{i,j}^m(x,y) < u_i^j(z), \frac{\partial u_i^k(z)}{\partial z} > + \Sigma_i \varphi_{i,k}^m(x,y) < u_i^k(z), u_i^k(z) > \\ &= Q_{i,k}^m < u_i^k(z), u_i^k(z) > -\xi_m \varphi_i^{m,+}(x,y) u_i^k(z_i^+) \quad (k = 1, \cdots, N)\end{aligned} \tag{13}$$

where $\varphi_i^{m,+}(x,y)$ is the incoming angular flux at the top of the i-th mesh. In order to compute Eq. (12) and Eq. (13), the following linear basis functions are introduced:

$$u_i^1(z) = 1, \;\; z \in [z_i^-, z_i^+] \;, \tag{14}$$

$$u_i^2(z) = \frac{2}{\Delta_i}\left(z - \frac{z_i^- + z_i^+}{2}\right), \; z \in [z_i^-, z_i^+] \;, \tag{15}$$

where Δ_i is the axial height of the i-th mesh. By inserting Eqs. (14) and (15) into Eqs. (12) and (13), the resulting equations can be written in a matrix-vector notation as:

$$\begin{aligned}&\begin{bmatrix} \Omega_R \bullet \nabla & 0 \\ 0 & \Omega_R \bullet \nabla \end{bmatrix} \begin{bmatrix} \varphi_{i,1}^m(x,y) \\ \varphi_{i,2}^m(x,y) \end{bmatrix} + \begin{bmatrix} \Sigma_i + \xi_m / \Delta_i & \xi_m / \Delta_i \\ -3\xi_m / \Delta_i & \Sigma_i + 3\xi_m / \Delta_i \end{bmatrix} \begin{bmatrix} \varphi_{i,1}^m(x,y) \\ \varphi_{i,2}^m(x,y) \end{bmatrix} \\ &= \begin{bmatrix} Q_{i,1}^m + \xi_m \varphi_i^{m,-}(x,y) / \Delta \\ Q_{i,2}^m - 3\xi_m \varphi_i^{m,-}(x,y) / \Delta \end{bmatrix}, \;\; \text{for } \xi_m > 0\end{aligned} \tag{16}$$

$$\begin{aligned}&\begin{bmatrix} \Omega_R \bullet \nabla & 0 \\ 0 & \Omega_R \bullet \nabla \end{bmatrix} \begin{bmatrix} \varphi_{i,1}^m(x,y) \\ \varphi_{i,2}^m(x,y) \end{bmatrix} + \begin{bmatrix} \Sigma_i - \xi_m / \Delta_i & \xi_m / \Delta_i \\ -3\xi_m / \Delta_i & \Sigma_i - 3\xi_m / \Delta_i \end{bmatrix} \begin{bmatrix} \varphi_{i,1}^m(x,y) \\ \varphi_{i,2}^m(x,y) \end{bmatrix} \\ &= \begin{bmatrix} Q_{i,1}^m - \xi_m \varphi_i^{m,-}(x,y) / \Delta \\ Q_{i,2}^m - 3\xi_m \varphi_i^{m,-}(x,y) / \Delta \end{bmatrix}, \;\; \text{for } \xi_m < 0\end{aligned} \tag{17}$$

Eqs. (16) and (17) can be concisely expressed as a single form:

$$\Omega_R^m \cdot \nabla \Psi_i^m(\mathrm{x},\mathrm{y}) + \boldsymbol{\Sigma}_i^m \Psi_i^m(\mathrm{x},\mathrm{y}) = S_i^m \;, \tag{18}$$

where $\Psi_i^m(\mathrm{x},\mathrm{y})$ is the angular flux vector, which consists of the $\varphi_{i,1}^m(x,y)$ and $\varphi_{i,2}^m(x,y)$. The cross section matrix, $\boldsymbol{\Sigma}_i^m$, and the source vector, S_i^m, can be defined as:

$$\boldsymbol{\Sigma}_i^m = \begin{bmatrix} \Sigma_i + \xi_m / \Delta_i & \xi_m / \Delta_i \\ -3\xi_m / \Delta_i & \Sigma_i + 3\xi_m / \Delta_i \end{bmatrix}, \quad S_i^m = \begin{bmatrix} Q_{i,1}^m + \xi_m \varphi_i^{m,-} / \Delta \\ Q_{i,2}^m - 3\xi_m \varphi_i^{m,-} / \Delta \end{bmatrix} \quad \text{for } \xi_m > 0 \tag{19}$$

$$\boldsymbol{\Sigma}_i^m = \begin{bmatrix} \Sigma_i - \xi_m / \Delta_i & \xi_m / \Delta_i \\ -3\xi_m / \Delta_i & \Sigma_i - 3\xi_m / \Delta_i \end{bmatrix}, \quad S_i^m = \begin{bmatrix} Q_{i,1}^m - \xi_m \varphi_i^{m,+} / \Delta \\ Q_{i,2}^m - 3\xi_m \varphi_i^{m,+} / \Delta \end{bmatrix} \quad \text{for } \xi_m < 0 \tag{20}$$

Eq. (18) is analogous to the conventional 2D MOC equation. By introducing the flat source approximation and following the 2D MOC solution scheme, it can be solved analytically along characteristic lines as:

$$\Psi_i^m(s) = \Psi_i^m(0)\exp(-\Sigma_i^m s/\sin\theta_m) + (\Sigma_i^m)^{-1}(1-\exp(-\Sigma_i^m s/\sin\theta_m))\mathrm{S}_i^m \ , \tag{21}$$

where s is the segment length of the characteristic line passing the i-th mesh and θ_m is the polar angle of m-th discretized angle. The segment averaged angular flux is computed as:

$$\Psi_{i,avg}^m = (\Sigma_i^m)^{-1}(\Psi_{i,\mathrm{in}}^m - \Psi_{i,\mathrm{out}}^m)\frac{\sin\theta_m}{s} + (\Sigma_i^m)^{-1}S_i^m \ . \tag{22}$$

In order to solve the MOC equation in an explicit manner, the incoming angular flux at the interface needs to be known. If a 3D geometry is constructed in such a way that a 2D planar geometry is extruded axially and the resulting 3D geometry is divided into several "planes," the incoming angular flux at the bottom of each mesh in each "plane" can be explicitly computed based on either the lower plane solution or the boundary condition. Therefore, the 3D transport problem appears similar to a series of 2D planar problems defined by Eq. (18) that are coupled together via element-wise boundary conditions on the incident surface. The coupled 2D problems are most efficiently solved using a sweeping scheme consistent with the direction of travel.

This method is very similar to the 2D/1D approach but the synthesis of 2D and 1D solutions that causes the numerical instability problem is eliminated. In reality, the MOCEX methodology is a true 3D methodology as the characteristic lines that are placed in the radial "planes" each have an axial basis function expansion within each radial element. In the current implementation, we average this solution over a given element for a given direction to simplify the memory storage and reduce the computational burden. It is not clear at this point what accuracy issues this causes as the approach is conservative and effectively acts as a slight smearing of the flux over a rather small area. It is important to note that there is no smearing of flux data in the radial plane, only the axial direction.

Anisotropic Scattering Treatment

Because fast spectrum systems require higher order anisotropic scattering treatments, the test code was updated to handle P_3 scattering orders where P_1 anisotropic scattering is the default option. At this point we have only tested the P_1 anisotropic scattering and need further study to verify the P_3 scattering capability. The conventional isotropic scattering with transport correction can be used also.

Parallelization

In order to reduce the computational time in the 2D MOC and MOCEX calculations, a parallel computation capability was implemented. Since a unit cell calculation is relatively small, spatial domain decomposition is not applicable. Consequently, angular decomposition was used in addition to including a multi-threading calculation capability using OpenMP.

2.3.2 Thermal Cross Sections

In the Dancoff approach, the resonance table as a function of background and temperature for each isotope is generated from NJOY. The smooth cross sections and the intermediate resonance (IR) parameters are prepared as well. The fixed source problem with the source ($\lambda\Sigma_p$) is solved to determine the Dancoff factors for fuel and cladding regions.

$$RI_t\,(E)=\Sigma_t^f\,(E)\frac{\Sigma_s^f+\Sigma_e}{\Sigma_t^f\,(E)+\Sigma_e}\frac{1}{E}, \tag{23}$$

where RI_t = total resonance integral that is calculated by solving the fixed source problem, $\Sigma_t^f\,(E)$ = total cross section for fuel, Σ_s^f = scattering cross section for fuel, Σ_e = escape cross section defined by the inverse of the average chord length, $1/\overline{\ell}=S/4V$. Assuming a black pin (i.e., $\Sigma_t^f\ \rightarrow\infty$), Eq. (23) becomes

$$RI_t\,(E)=\Sigma_s^f+\Sigma_e=\Sigma_b+D\Sigma_e \ \ \rightarrow\ \ D=(RI_t\,(E)-\Sigma_b)/\Sigma_e\,, \tag{24}$$

where Σ_b = background cross sections for homogeneous mixture and D = Dancoff factor. Once the total resonance integral is solved using the FSP assuming the infinite cross sections in the fuel region, the Dancoff factor for the fuel region can be calculated using Eq. (24). When resonant isotopes are included in the cladding region, the similar approach can be applied to calculation the Dancoff factor for the cladding region.

Since the one-term rational approximation is known to be inaccurate, the non-escape probability (P_{ff}) can be represented by the two-term rational approximation in Eq. (25). Finally, the resonance cross section (σ_x^i) for the reaction x of the resonant isotope i is determined by Eq. (26).

$$P_{ff}\,(E)=\sum_{n=1}^{2}\frac{\beta_n\Sigma_t\,(E)}{\Sigma_t\,(E)+\alpha_n\Sigma_e}, \tag{25}$$

$$\sigma_x^i=\frac{\sum_{n=1}^{2}\beta_n RI_{xn}^i\,(\sigma_{bn}^i)}{1-\sum_{n=1}^{2}\dfrac{\beta_n RI_{an}^i\,(\sigma_{bn}^i)}{\sigma_{bn}^i}}, \tag{26}$$

where RI_x = resonance integral of reaction x , σ_{bn}^i = $\left(\sum_k \lambda\Sigma_p^k+\alpha_n\Sigma_e\right)/N^i$, and α_n,β_n = n-th term coefficients.

The cross section generation option using the preceding Dancoff approach was implemented into MC^2-3, along with the additional cross section library for the Dancoff approach and the thermal energy range up to 10^{-5} eV. When the thermal cross section option is called, the cross sections below 10 eV are generated using the new approach and library. Currently two group structures, 73 and 894 groups, were made available. The 894 group structure was made based on the existing 1041 group structure, the last 190 groups (below 10 eV) of which are replaced by 43 thermal groups from the new approach and library. Similarly, the 73 group structure was created based upon the existing 70 group structure where the last 40 groups are replaced by 43 thermal groups. Figure 14 shows the new 73 and 894 group structures in terms of lethargy. As can be seen, the existing library covers up to 0.414 eV even though the new library contains cross sections up to 10^{-5} eV.

Preliminary verification tests were performed for the new library using homogeneous compositions of typical LWR, HWR, and SFR. Isotopes and number densities for those compositions are listed in the Table 4, Table 3, and Table 5. The k-infinitive solutions of MC^2-3 were compared with MCNP solutions, indicating that k-infinitive values for all cases are in good agreement between MC^2-3 and MCNP. However, relatively lager differences are observed in the HTR composition cases. This may be because the new thermal cross sections have only 43 groups in the thermal energy range below 10 eV. Therefore, the solution accuracy would be improved by allocating more energy groups.

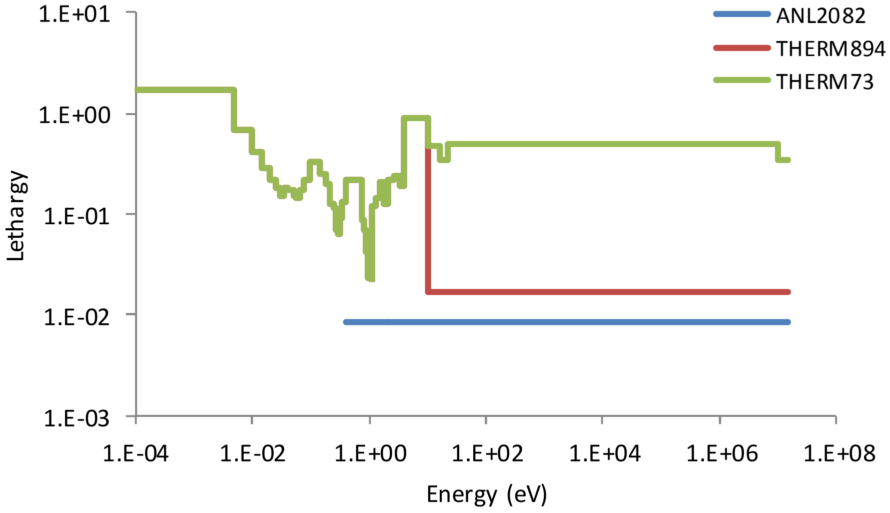


Figure 14. New Group Structures for Thermal Cross Sections in MC^2-3.

Table 3. Isotopes and Number Densities of LWR Compositions

Case	1		2		3		4		5		6	
Isotope & Number Density	U-234	2.03E-06	U-234	2.03E-06	U-234	2.03E-06	U-234	2.03E-06	U-234	2.03E-06	U-234	2.03E-06
	U-235	2.38E-04	U-235	2.38E-04	U-235	2.38E-04	U-235	2.38E-05	U-235	2.38E-04	U-235	2.38E-06
	U-236	1.10E-06	U-236	1.10E-06	U-236	1.10E-06	U-236	1.10E-06	U-236	1.10E-06	U-236	1.10E-06
	U-238	7.36E-03	U-238	7.36E-04	U-238	7.36E-05	U-238	7.36E-03	U-238	7.36E-06	U-238	7.36E-03
	O-16	2.89E-02	O-16	2.89E-02	O-16	2.89E-02	O-16	2.89E-02	O-16	2.89E-02	O-16	2.89E-02
	Zr-90	4.28E-03	Zr-90	4.28E-03	Zr-90	4.28E-03	Zr-90	4.28E-03	Zr-90	4.28E-03	Zr-90	4.28E-03
	H-1	2.75E-02	H-1	2.75E-02	H-1	2.75E-02	H-1	2.75E-02	H-1	2.75E-02	H-1	2.75E-02

Table 4. Isotopes and Number Densities of SFR Compositions

Case	1		2		3		4		5		6		7	
Isotope &	U-235	1.11E-05	U-235	7.00E-04	U-235	1.44E-05	U-238	5.00E-03	U-238	7.00E-03	U-238	6.00E-03	U-235	1.59E-05
Number	U-238	5.49E-03	U-238	5.00E-03	U-236	9.20E-07	Pu-239	3.00E-03	Pu-239	1.00E-03	Pu-239	2.00E-03	U-238	7.53E-03
Density	Pu-239	9.11E-04	Pu-239	9.11E-04	U-238	9.05E-03	O-16	2.00E-02	O-16	1.60E-02	O-16	1.60E-02	Cr-50	1.67E-04
	Pu-240	3.52E-04	Fe-56	4.00E-01	Pu-239	1.57E-03	Fe-56	8.00E-03	Fe-56	2.00E-02	Fe-56	2.00E-02	Fe-56	1.26E-02
	Pu-241	1.03E-04			Pu-240	1.67E-04	Na-23	1.60E-02	Na-23	8.00E-03	Na-23	8.00E-03	Ni-58	1.84E-03
	Pu-242	5.79E-05			Pu-241	1.10E-05							Mo-92	4.54E-05
	Cr-50	3.22E-03			Pu-242	7.85E-07							Na-23	9.44E-03
	Fe-56	1.26E-02			Cr-50	1.05E-04							Mn-55	3.64E-04
	Ni-58	1.84E-03			Fe-56	1.49E-02							O-16	1.51E-02
	Mo-92	4.54E-05			Ni-58	6.80E-05								
	Na-23	9.44E-03			Mo-92	8.17E-05								
	O-16	1.37E-02			Na-23	7.15E-03								
					Mn-55	1.07E-04								
					Zr-90	1.68E-03								

Table 5. Isotopes and Number Densities of HTR Compositions

Case	1		2		3		4		5	
Isotope &	U-235	1.064E-04	U-235	1.064E-04	U-235	1.064E-04	U-235	1.064E-04	U-235	1.064E-04
Number	U-238	6.453E-04	U-238	3.226E-04	U-238	6.453E-05	U-238	1.291E-05	U-238	1.291E-05
Density	O-16	1.128E-03	O-16	1.128E-03	O-16	1.128E-03	O-16	1.128E-02	O-16	2.255E-02
	C-12	6.063E-02	C-12	6.063E-02	C-12	6.063E-02	C-12	6.063E-01	C-12	1.213E+00
	SI-28	2.516E-03	SI-28	2.516E-03	SI-28	2.516E-03	SI-28	2.516E-02	SI-28	5.032E-02

Table 6. k-infinitive Comparison between MC^2-3 and MCNP for Various Homogeneous Compositions

SFR			LWR			HTR		
Case	MCNP	MC^2-3	Case	MCNP	MC^2-3	Case	MCNP	MC^2-3
1	1.35790	135	1	1.28101	77	1	1.06357	192
2	1.09577	-105	2	1.70407	55	2	1.24147	-103
3	1.63854	7	3	1.83689	37	3	1.5387	-242
4	2.06898	64	4	0.58602	50	4	1.84938	-82
5	1.24101	150	5	1.87309	115	5	1.73382	-232
6	1.75029	108	6	0.15768	76			
7	0.18881	72						

2.3.3 Gamma Library

The new gamma libraries are based on 2,082 neutron and 94 gamma groups. The group structures were selected based on the MC^2-3 neutron ultrafine group structure and the 94 gamma-group structure of the Cross Section Evaluation Working Group (CSEWG) [13]. The 94-group structure covers the energy range from 5 keV to 20 MeV, and it is considered sufficiently fine since gamma cross sections are smooth in most of the energy range important for gamma heating calculations. Figure 15 digests the code system and procedure used in generating the new gamma libraries.

The NJOY code is used to generate the prompt gamma production matrix and the interaction cross section library as well as the neutron and gamma heating factors at the infinite dilute condition. NJOY input files and process file are generated using the newly developed utility code named PMCS (Program for MC^2-3 Sub-library). The prompt gamma production matrices are generated using the RECONR, BROADR, HEATR and GROUPR modules of NJOY and the gamma interaction cross sections are generated using the RECONR and GAMINR modules of NJOY and the 1/E weighting spectrum.

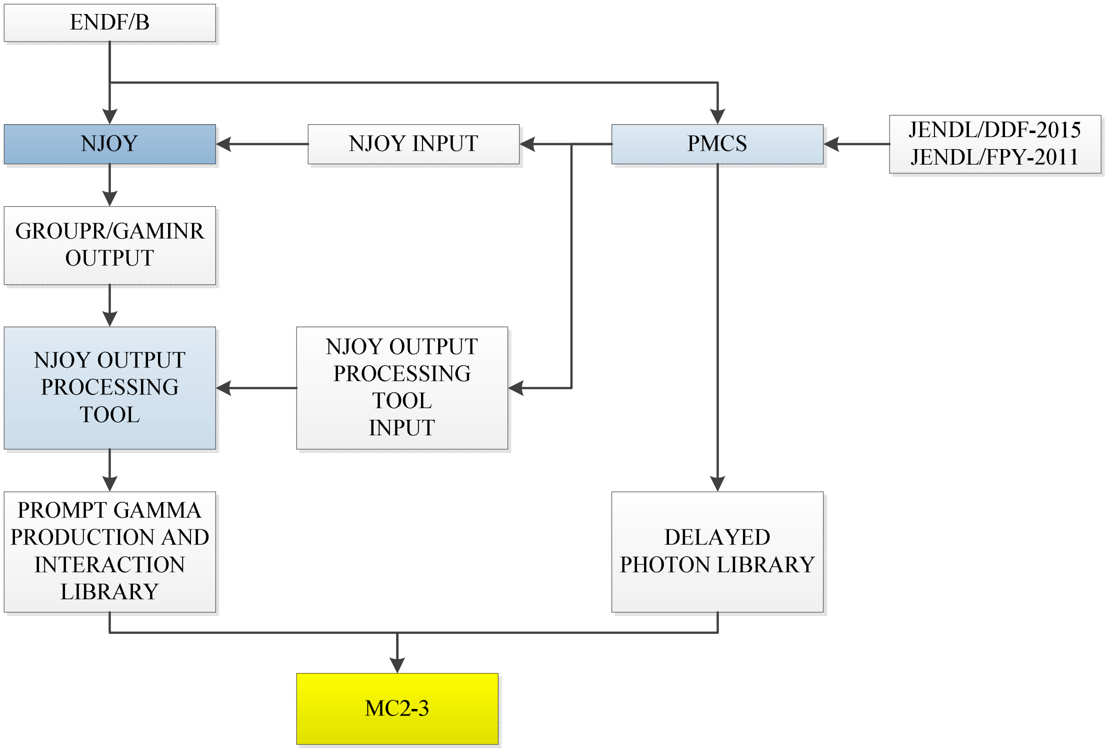

ENDF/B
NJOY
NJOY INPUT
PMCS
JENDL/DDF-2015
JENDL/FPY-2011
GROUPR/GAMINR OUTPUT
NJOY OUTPUT PROCESSING TOOL
NJOY OUTPUT PROCESSING TOOL INPUT
PROMPT GAMMA PRODUCTION AND INTERACTION LIBRARY
DELAYED PHOTON LIBRARY
MC2-3

Figure 15. Calculation Flow for Gamma Library of MC^2-3.

The NJOY output-processing tool converts the output files of NJOY (GENDF format) into the prompt gamma production yield and interaction cross section libraries of MC^2-3. To account for the self-shielding effects on gamma production, the production matrices for capture and fission reactions are converted into the yield matrices, while the production matrices are directly used for non-elastic neutron scattering reactions. The self-shielding effects on the KERMA factor of neutron reactions is accounted for in terms of partial kinetic energy release per reaction, which is determined by dividing the partial KERMA factor by the neutron cross section and storing it in the gamma yield library. Later, the KERMA factors for given compositions are calculated in MC^2-3 by multiplying these partial kinetic energy releases per reaction with self-shielded cross sections. For gamma interactions, the KERMA factors obtained from NJOY are directly used.

PMCS also calculates the delayed gamma production matrix resulting from fission product decays and the heating contribution of delayed betas. Assuming that the delayed beta energy is deposited locally without beta transport in the heating calculation, the kinetic energy of delayed betas is retrieved from the ENDF/B File 1. To calculate the delayed gamma production matrix, PMCS constructs the decay chains of fission products by reading the fission yield and radioactive decay sub-libraries. The gamma production matrix is then constructed by aggregating all of the decay gamma spectra. In this work, the JENDL decay (JENDL/DDF-2015) and fission yield (JENDL/FPY-2011) data files [14] were used to generate the delayed gamma yield matrices. It is noted that the JENDL decay data was compiled in 2011 with the fission product decay data file in order to make the average beta and gamma decay energy with their spectral data using the theoretically calculated spectra and the total absorption gamma-ray spectroscopy (TAGS) data for the nuclides with incomplete decay scheme [14,15]. Table 7 compares the delayed gamma energy per fission estimated by PMCS using the fission yield and decay data of ENDF/B-VII.1 and JENDL with the delayed gamma energy per fission in the File 1, MT 458 data of ENDF/B-VII.1. It can be seen that the delayed gamma energies estimated with JENDL libraries agree with the MT 458 data better than ENDF/B-VII.1 libraries. In the PMCS calculation, the delayed gamma production matrices are renormalized such that the total delayed gamma energy is equal to the reference value of MT 458 data.

Table 7. Comparison of Delayed Gamma Energy per Fission (MeV) Estimated from Fission Product Decays with MT 458 Data for Major Actinides

Fission Yield and Decay Libraries	^{235}U	^{238}U	^{239}Pu	^{240}Pu	^{241}Pu	^{242}Pu
ENDF/B VII.1	5.62	6.87	4.56	5.00	5.43	5.81
JENDL	6.22	7.72	5.25	5.77	6.27	6.65
Reference (MT 458 of File 1)	6.33	8.25	5.17	6.49	6.40	6.82

* Incident neutron energy = 0.0253 (eV)

The gamma flux spectrum is determined by solving the multigroup gamma transport equation

$$\Omega \cdot \nabla \phi_\gamma^g + \Sigma_t^g \phi_\gamma^g = \sum_{g'} \Sigma_{\gamma\gamma}^{g' \to g} \phi_\gamma^{g'} + \sum_n \Sigma_{\gamma n}^{n \to g} \phi_n^n + S_\gamma^g , \qquad (27)$$

where the subscript n and γ denotes the neutron and gamma quantities, respectively, and the superscript n and g represent the neutron and gamma groups, respectively. That is, ϕ_n^n is the n-th group neutron flux determined from the neutron transport calculation and ϕ_γ^g is the g-th group gamma flux. Σ_t^g is the macroscopic total cross section of gamma group g, $\Sigma_{\gamma\gamma}^{g' \to g}$ is the macroscopic scattering cross section from gamma group g' to g, $\Sigma_{\gamma n}^{n \to g}$ is the macroscopic production cross section from neutron group n to gamma group g, and S_γ^g represents the external gamma source. The gamma transport calculation can be performed for a homogeneous medium or a one-dimensional (1D) slab or cylindrical unit cell problem.

The calculated gamma group flux is used in generating the broad-group gamma interaction cross sections and heating factors. For each broad group, the heating factor and interaction cross sections are determined by collapsing the 94-group cross sections in the library as

$$\sigma_x^G = \frac{\sum_{g\in G}\sigma_x^g\phi_\gamma^g}{\sum_{g\in G}\phi_\gamma^g},\quad \sigma_x^{G'\to G} = \frac{\sum_{g\in G}\sum_{g'\in G'}\sigma_x^{g'\to g}\phi_\gamma^{g'}}{\sum_{g'\in G'}\phi_\gamma^{g'}}, \qquad (28)$$

$$K_x^G = \frac{\sum_{g\in G}K_x^g\phi_\gamma^g}{\sum_{g\in G}\phi_\gamma^g}, \qquad (29)$$

where σ and *K* represent the interaction cross section and gamma KERMA factor, respectively. The subscript *x* denotes the reaction type, and the superscript *g* and *G* denoting the fine and broad gamma group indices, respectively.

As the first step to test the MC^2-3 capability for gamma data generation, the gamma flux distribution and the gamma interaction cross sections were calculated for a cylindrical fuel pin problem of ABTR and compared with the MCNP6 Monte Carlo solution. Both calculations were performed with the cross section libraries based on the ENDF/B VII.0 data. The MCNP6 calculation was performed with 1,000 active cycles and 10,000 histories per cycle. The delayed fission gammas were not considered since the MCNP6 cross section library lacks the corresponding data. The relative statistical errors of MCNP6 results were less than 0.3% except for the high-energy region greater than 10 MeV.

Figure 16 shows the 94-group gamma flux distributions in the fuel region obtained from MC^2-3 calculations with and without delayed gammas and that from MCNP6 calculation without delayed gammas.

It is clearly seen that the gamma flux distribution of MC^2-3 without delayed gammas agrees very well with the MCNP6 solution. Note that the sharp peak at 0.5 MeV is mainly due to the gamma rays from the neutron capture and fission reactions of ^{235}U and the pair-production reactions in the high-energy range. The peak at 0.8 MeV is mainly due to the inelastic scattering reactions of ^{56}Fe.

Figure 17 compares the 94- and 21-group total gamma interaction cross sections calculated with MC^2-3 and MCNP6 for uranium and iron, respectively. It can be seen the 21-group gamma interaction cross sections determined with MC^2-3 agree very well with those determined with MCNP6 except for the groups 17 to 19 (30 – 100 keV) and group 21 (5 – 20 keV). These differences in low-energy cross sections result from the differences in the 94-group gamma libraries themselves, which are caused by the abrupt changes in photoelectric absorption cross sections due to the absorption edges of different electron shells.

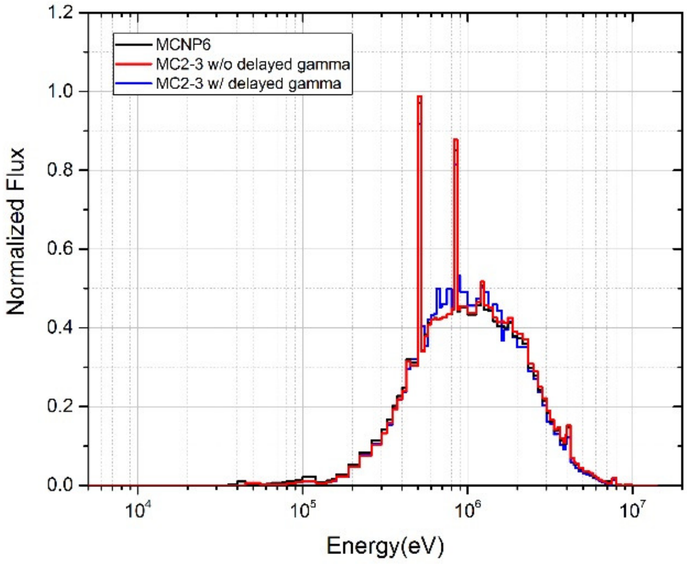


Figure 16. 94-group Gamma Flux Distributions in ABTR Fuel.

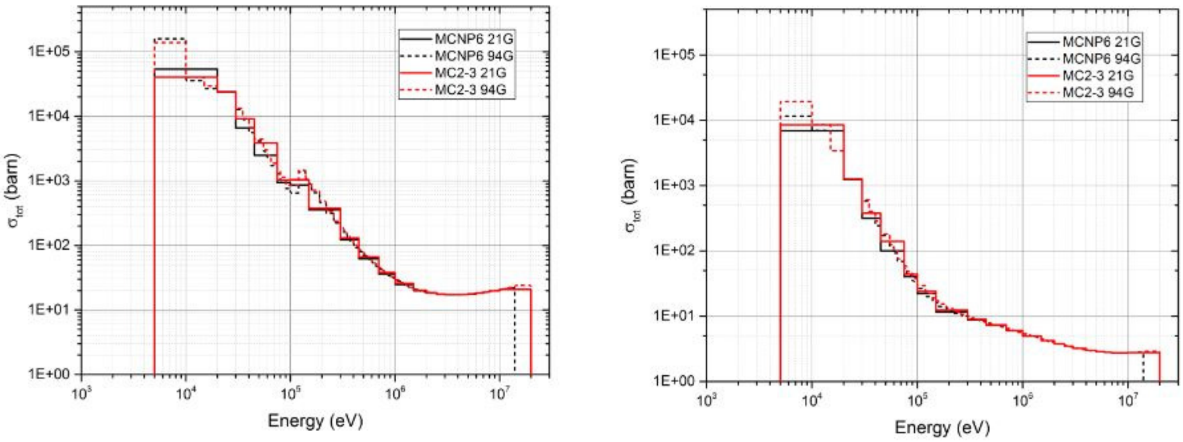


Figure 17. Total Gamma Cross Section of Uranium (left) and Iron (right) in ABTR Fuel Pin.

These results indicate that the 94-group structure of CSEWG is not sufficiently fine to represent the photoelectric absorption edges accurately. However, as can be seen in Figure 17, the flux level is very low in the energy range below 100 keV, where the photoelectric absorption cross sections are important. Therefore, it is judged that the 94-group structure is adequate for the coupled neutron and gamma heating calculations.

In order to examine the effects on power density, coupled neutron and gamma heating calculations were performed for the EBR-II benchmark problem [16] using the DIF3D-VARIANT nodal transport code [17] (noted as VARIANT hereafter) with the cross sections prepared with the new gamma data generation scheme of MC^2-3. The EBR-II benchmark problem is based on the core configuration of Run 132B, which consists of 71 regular driver fuel assemblies, 13 half-worth fuel driver assemblies, 10 control and safety assemblies, six structural assemblies, and six instrumented assemblies. The detailed core specifications can be found in References [16] and [18]. Neutron and gamma cross section library was generated using the ENDF/B-VII.0 data. For comparison, a MCNP6 calculation was performed with 180 active cycles with 100,000 histories per cycle. The delayed fission gammas were not considered in both MC^2-3/VARIANT and MCNP6 calculations for a consistent comparison. The results of MCNP6 have relative statistical errors less than 0.2%.

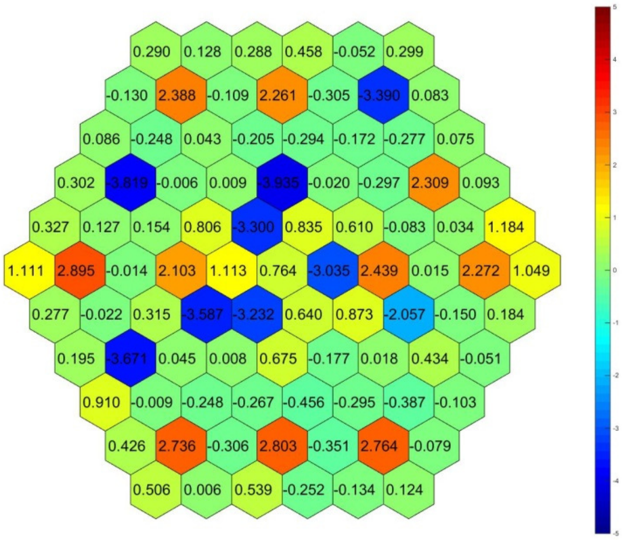


Figure 18. Relative Difference (%) in Assembly Power Density between MCNP6 and MC^2-3/VARIANT (with Current 21-group Gamma Library).

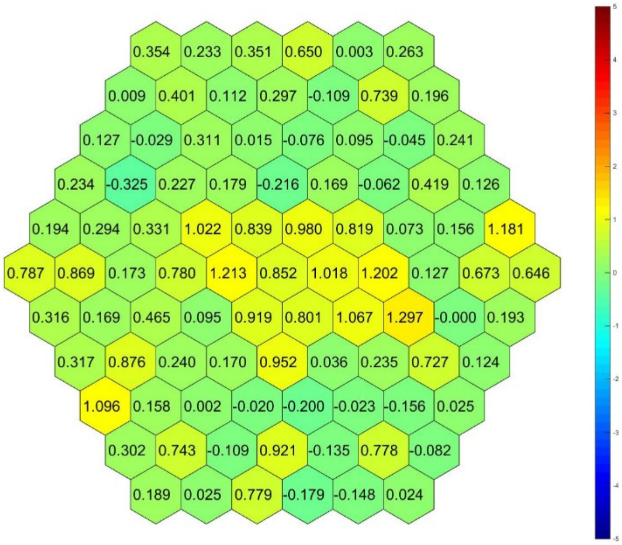


Figure 19. Relative Difference (%) in Assembly Power Density between MCNP6 and MC^2-3/VARIANT (with New Gamma Data Generation Scheme of MC^2-3).

The VARIANT assembly powers obtained with the current 21-group gamma library of MC^2-3 [3] agreed with the MCNP solutions within 4%. Figure 18 shows the relative differences in assembly power of the MC^2-3/VARIANT solution from the MCNP6 solution. It is noted that the MC^2-3/VARIANT solution underestimates the power densities in structural assemblies but overestimates those in the control and safety assemblies.

Figure 19 shows the relative differences in assembly power of the VARIANT solution obtained with the new gamma data generation scheme of MC^2-3 from the MCNP6 solution. A set of 21-group gamma production and interaction cross sections were generated by MC^2-3 calculations with the new 94-group gamma yield and interaction cross sections. As shown in Figure 19, the differences in assembly power from the MCNP6 solution are reduced noticeably to less than 1.3% by employing the new gamma library and gamma transport module of MC^2-3.

2.4 Mesh Utilities

The neutronics toolset includes utility codes to facilitate finite element meshing necessary for the PROTEUS neutronics solvers. Several updates and additions were made to the mesh utility codes, including improving the ability to merge smaller meshes into a larger mesh, changing the finite element basis functions in the meshes, upgrading the Exodus format mesh converter to use newer versions of CUBIT, and developing a utility to deform 3D meshes according to user-specified strain functions. These utilities have already significantly streamlined neutronics analysis by providing the ability to quickly modify and merge finite element meshes after they have been generated.

2.4.1 Mesh Generation Toolkit

Two major changes were made to update and improve the mesh utility codes associated with PROTEUS. First, the merge mesh utility code was found to be too cumbersome to use and was generalized to include an axial merging capability. Second, the CUBIT derived meshes consisted of mixed Lagrangian and Serendipity element types which cannot be combined because of the different element types and orders and thus a mesh type modification routine was added.

The primary problem with the existing mesh merging utility is that is uses lattice based input system. This approach is great when building a hex or Cartesian lattice of similar meshes (i.e. assemblies), but for arbitrary sets of meshes the merge calculation is considerably complicated. To accommodate the ability to merge an arbitrary set of meshes, a new mesh file format (*.mergexyz) was defined (as opposed to *.merge) and implemented as part of the recursive setup routine of MeshLibrary in PROTEUS.

Table 8 gives the input commands in the new file format along with an example input. Table 9 gives the input commands for the previous *.merge option for comparison purposes. The purpose of the new merge option is identical to that of the preceding merge functionality but the new input indicates a change in its behavior. To begin, we point out that the new input allows the user to automatically merge the assignment files of different meshes together. With the merge input, the

RadialLattice utility code allowed the user to automatically rename the assignment and region information using a pin or assembly identifier. That approach was not deemed important for a generic merge operation and thus the user must define the region names and assignment information appropriately (i.e. to prevent merging of materials with similar names).

Another change in behavior is that the new file format allows one to shift meshes with respect to the axial direction indicating the ability to merge 3D meshes together. The actual merge function that is used by both *.merge and *.mergexyz allowed this operation and was already tested. In the radial lattice input of *.merge, one could only considered radial placements of meshes, and thus it was not appropriate to utilize the axial translation option. In the *.mergexyz input, this functionality is fully usable and one can trivially determine the proper offsets to use given a mesh generated by CUBIT, grid, or ufmesh.

Table 8. General Purpose Mesh Merge Control Input *.mergexyz

Input Keyword	Format	Purpose
OUTPUT_FILE	<new assignment output>	Specify the output file for writing assignment data
GEOMETRY_DATA	<X> <Y> <Z> <mesh> [input assignment]	Specify an input mesh and its X,Y,Z offsets along with optional assignment

Example

```
OUTPUT_FILE  ./erase.assignment
GEOMETRY_DATA 1.0 0.0 0.0 ./data/cart_1.nemesh ./data/cart_1.assignment
GEOMETRY_DATA 2.0 0.0 0.0 ./data/cart_2.nemesh ./data/cart_2.assignment
GEOMETRY_DATA 3.0 0.0 0.0 ./data/cart_3.nemesh ./data/cart_3.assignment
GEOMETRY_DATA 4.0 0.0 0.0 ./data/cart_4.nemesh ./data/cart_4.assignment
GEOMETRY_DATA 5.0 0.0 0.0 ./data/cart_5.nemesh ./data/cart_5.assignment
```

Table 9. Control Input for the Radial Lattice Code *.merge

Input Keyword	Format	Purpose
RADIAL_LATTICE	<type> <pitch> <x size> <y size> <Type> = <hexagonal> or <Cartesian>	Specify the type of grid, its pitch, and number of x and y dimensions
GEOMETRY_DATA	<alias> <input mesh> [input assignment]	Specify an input mesh and assignment file and assign them both an alias
OUTPUT_FILE	<new assignment output>	Specify the output file for writing assignment data
GRIDLINE_DATA	<y position> <x1> <x2> … <xN>	Define the axial extrusion

Example

```
RADIAL_LATTICE Hex 8.0  13 13
GEOMETRY_DATA A1 ./hex01.axial
GEOMETRY_DATA A2 ./hex02.axial
GEOMETRY_DATA A3 ./hex03.axial
```

Input Keyword	Format	Purpose

```
GEOMETRY_DATA A4 ./hex04.axial
GRIDLINE_DATA 13   O  A3  A3  A3  A3  A3   O   O   O   O   O   O   O
GRIDLINE_DATA 12  A3  A3  A3  A3  A3  A3  A3  A3   O   O   O   O   O
GRIDLINE_DATA 11  A3  A3  A4  A2  A2  A2  A4  A3  A3   O   O   O   O
GRIDLINE_DATA 10  A3  A3  A2  A2  A2  A2  A2  A2  A3  A3   O   O   O
GRIDLINE_DATA  9  A3  A3  A2  A2  A4  A1  A4  A2  A2  A3  A3   O   O
GRIDLINE_DATA  8  A3  A3  A2  A2  A1  A1  A1  A1  A2  A2  A3  A3   O
GRIDLINE_DATA  7   O  A3  A4  A2  A4  A1  A4  A1  A4  A2  A4  A3   O
GRIDLINE_DATA  6   O  A3  A3  A2  A2  A1  A1  A1  A1  A2  A2  A3  A3
GRIDLINE_DATA  5   O   O  A3  A3  A2  A2  A4  A1  A4  A2  A2  A3  A3
GRIDLINE_DATA  4   O   O   O  A3  A3  A2  A2  A2  A2  A2  A2  A3  A3
GRIDLINE_DATA  3   O   O   O   O  A3  A3  A4  A2  A2  A2  A4  A3  A3
GRIDLINE_DATA  2   O   O   O   O   O  A3  A3  A3  A3  A3  A3  A3  A3
GRIDLINE_DATA  1   O   O   O   O   O   O   O  A3  A3  A3  A3  A3   O
```

The second modification made to the mesh utilities was to add the ability to change the finite element basis type. This was done as a consequence of the CUBIT mesh converter producing 2D meshes with Lagrangian and serendipity element types. These meshes cannot be extruded as lagrangian triangles will extrude to 18 vertices per element and serendipity triangles will extrude to 15 vertices per element. The difference of course is that the serendipity elements do not have face centered vertices on the square surfaces. A Lagrangian triangular element adjacent to a serendipity quadrilateral is not an issue for two-dimensional meshes, but when one extrudes it, the missing vertices is not consistent with a conformal meshing based code such as PROTEUS-SN. To fix this, the mesh utility code MT_ChangeFEbasis.x was modified to use the input in Table 10.

Table 10. Command Line Options for the MT_ChangeFEbasis.x Mesh Tool

Command Line Option	Purpose
New FE Order	The new order of the FE basis functions
Input File	The mesh input file
Output File	The mesh output file
[Element Type]	The type of element to switch to: null, serendipity, lagrange, gausslobato, nonconform
[Mesh Check]	Optional check to verify the mesh transformation
Example: ./MT_ChangeFEbasis.x 2 input.ascii out.pntmesh NULL yes	

In Table 10, a new optional input called "Element type" was included which has specific allowed values. In this case the user can now modify a given mesh and consistently switch the element basis from serendipity to lagrangian which means that the existing basis functions are used to interpolation the vertex locations of the new element type. This approach exactly preserves the volume of the two elements with negligible changes to its surface contour. To allow users to utilize this functionality in line with other operations such as merge and reduce the number of stored component meshes, a new mesh file format was constructed called .SwitchType. Table 10 gives the keyword file format information for .SwitchType along with an example input.

Table 11. Command Line Options for the MT_ChangeFEbasis.x Mesh Tool

Input Keyword	Format	Purpose
INPUT_MESH	<the input mesh>	Specify the input mesh to load that is to be converted
TARGET_TYPE	<element type> [order]	Specify the new element type

Example

```
  INPUT_MESH  ./my.cubit.ascii
  TARGET_TYPE SERENDIPTY       1
! TARGET_TYPE  LAGRANGE        0  ! Will retain the existing mesh basis order
! TARGET_TYPE  GAUSSLOBATTO    2
! TARGET_TYPE  NONCONFORM      1
```

From the example in Table 11, one can see that the input is relatively straightforward requiring a mesh input file target (such as .nodal or .ascii). The other input is the specification of a new element type and optional element order change. The element type information is defined in the manual noting that non-conform does not technically work at this time in PROTEUS-SN. The mesh tools manual was updated to indicate the new features and is currently under review.

2.4.2 EXODUS Mesh Convertor Utility

The EXODUS to ASCII mesh converter, which converts Cubit (EXODUS format) meshes to PROTEUS native format, was verified to work with Cubit version 15 when compiled with Intel 13.1. Cubit version 15 was needed to access ACIS geometries for TREAT created with the MCNP to ACIS converter. Additionally, verification tests for the EXODUS to ASCII converter were added to the nightly regression test suite.

2.4.3 Mesh Deformation Utility

Since a primary target use case of PROTEUS-SN is to directly analyze deformed geometry configuration (for instance, to calculate the reactivity feedback effect), we created a utility to apply deformations to a given mesh. Previously, the only way to obtain a deformed mesh was to run a fully coupled calculation through the SHARP toolkit with PROTEUS-SN, Nek5000, and the DIABLO structural mechanics code (requiring extensive inputs and neutronics, T/H, and structural mechanics user expertise). This is not convenient for quick testing and demonstration purposes. In order to quickly generate sample deformed meshes for testing purposes in PROTEUS-SN, the new mesh utility deforms any 3D PROTEUS-SN mesh in the X-, Y- and Z-dimension around a pinned coordinate.

Table 12. Input File Format for the MT_DeformMesh.x Mesh Tool

Input Keyword	Format	Description
SOURCE_MESHFILE	Path to file	Input mesh
VOLUME_FILE (Optional)	Path to file	Output file containing volume ratios of original and deformed mesh. Default is deformed_volumes.out.
PINNED_COORD_X (Optional)	Real value x_c	X-coordinate value of the pinned coordinate location. Default value is the average of the global maximum and minimum X-coordinates, $x_c = (x_{\min} + x_{\max})/2$.
PINNED_COORD_Y (Optional)	Real value y_c	Y-coordinate value of the pinned coordinate location. Default value is the average of the global maximum and minimum Y-coordinates, $y_c = (y_{\min} + y_{\max})/2$.
PINNED_COORD_Z (Optional)	Real value z_c	Z-coordinate value of the pinned coordinate location. Default value is the global minimum Z-coordinate, $z_c = z_{\min}$.
ZSTRAIN	Real value ε_z	Constant strain in the z-direction. Ex: ZSTRAIN = 0.02 indicates 2% expansion in the Z-direction from the pinned coordinate. Assign 0 for no expansion in z-direction. Ex: ZSTRAIN = -0.02 indicates 2% contraction.
Z_ARRAY	Real values $z_1 \ z_2 \ \dots \ z_N$	Space delimited list of Z-coordinates that spans the entire range of the mesh. It is recommended that the original mesh has axial planes at these Z-coordinates to avoid unexpected results.
XSTRAIN_ARRAY	Real values $\varepsilon_{z1}^x \ \varepsilon_{z2}^x \ \dots \ \varepsilon_{zN}^x$	Space delimited list of strains in the X-direction corresponding to each entry in Z_ARRAY. Strains are linearly interpolated between these values.
YSTRAIN_ARRAY	Real values $\varepsilon_{z1}^y \ \varepsilon_{z2}^y \ \dots \ \varepsilon_{zN}^y$	Space delimited list of strains in the Y-direction corresponding to each entry in Z_ARRAY. Strains are linearly interpolated between these values.
SKIPSOLVE (Optional)	YES/NO	YES tells the utility to read in the mesh and input file, print the mesh extents including default pinned coordinate, and quit. NO is the default for normal operation.

Example:

```
SOURCE_MESHFILE  ../base.ascii
VOLUME_FILE      deformed_volumes.out
! PINNED_COORD_X  center of bottom plane
! PINNED_COORD_Y  center of bottom plane
! PINNED_COORD_Z  center of bottom plane
ZSTRAIN       0.01
Z_ARRAY        0.0   110.54    169.63     194.95     200.0     280.36   330.61   345.68
XSTRAIN_ARRAY  0.0  -6.81E-02  -1.36E-01 -6.81E-02  -3.40E-02  3.40E-01 5.45E-01 6.81E-01
YSTRAIN_ARRAY  0.0  -6.81E-02  -1.36E-01 -6.81E-02  -3.40E-02  3.40E-01 5.45E-01 6.81E-01
```

The mesh deformation utility requires an input 3D mesh and a control input file (see Table 12) describing the strain functions to be applied in the X-, Y-, and Z- dimensions. Piecewise linear functions of Z are permitted for the X-strain and Y-strain functions; these functions must be specified by tabulated values at different Z-elevations. The X-strain and Y-Strain values represent radial expansion coefficients at different elevations. Additionally, a constant Z-strain (axial expansion) can be specified. Optionally, a pinned (X,Y,Z) coordinate about which to expand/contract the mesh can be specified. Otherwise the global (X,Y) midpoint at the lowest Z-plane is chosen to be the pinned coordinate by default.

This new mesh deformation tool therefore can create simple deformations where all assemblies are assumed to shrink/expand together in the same shape. While in reality assemblies should be permitted to deform according to unique axial shapes (outer vs. inner, fuel vs. reflector, etc.), this tool provides an approximate deformation scheme for testing purposes. An accompanying manual describing the control input file for this utility was created. Figure 20 shows an example input and output deformed mesh where the deformation is highly exaggerated for visualization purposes.

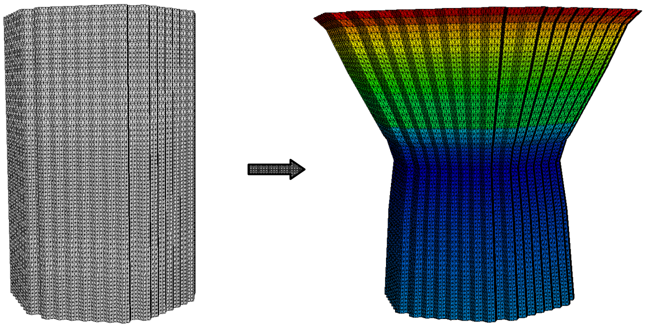


Figure 20. Example Deformed Mesh Generated by the New Mesh Deformation Utility.

In addition to deforming the mesh, a convenient output is printed with the block name and new atom densities for each block. This does not consider any special materials isotopic concentration changes, but can be directly copied into the material assignment file over the old density assignment, if the user wishes to perform the simplest calculation type. However, densities can be alternatively processed directly in PROTEUS-SN using the online density computation option mentioned in the previous section.

In the future we plan to create a utility which can perform more complex and realistic deformation, in particular the ability to perform grid plate expansion separate from bowing. This would allow, for example, the ducts to bow without being expanded (maintaining the same volume, but increasing/decreasing the distance between duct walls). However, creation of such a utility requires far more geometry information than currently required and provided by PROTEUS's mesh format and data structure. There is no concept of reactor geometry in the current mesh format as it is not needed for any mesh processing purposes. However, the UFmesh format developed last year can be expanded to allow definition and storage of reactor geometry (i.e. duct, pin, coolant, solid, liquid).

2.5 Alternative Cross Section Generation

The GenISOTXS code was developed in FY15 to process the Serpent-generated multigroup cross section in the ISOTXS format in order to provide alternative cross section generation methods for users. Earlier versions of Serpent did not provide the complete set of cross section data information needed for exact conversion to the ISOTXS format. The recent version of Serpent includes a more complete set of cross section information. However, Serpent mainly provides macroscopic cross sections, and even though it has an additional option to produce microscopic cross sections, isotopic scattering matrix data are still not produced. Responding to the strong need of users for isotopic cross sections, we added the OpenMC Monte Carlo code to this alternative cross section generation option.

Even though the OpenMC code is already able to generate multigroup cross sections, those cross sections have not yet been used for actual diffusion/transport codes like DIF3D. Therefore, many iterations were required with the OpenMC developers to complete its cross section generation capability by fixing bugs and adding new features required for correct cross sections.

```
for cell in TALLY_REGIONS:
 List_Tallys_tmp=[0]*9
 List_Tallys_tmp[0] = mgxs.TransportXS(domain=cell, groups=egroups,by_nuclide=True)
 List_Tallys_tmp[1] = mgxs.TotalXS(domain=cell, groups=egroups, by_nuclide=True)
 List_Tallys_tmp[2] = mgxs.CaptureXS(domain=cell, groups=egroups,by_nuclide=True)
 List_Tallys_tmp[3] = mgxs.AbsorptionXS(domain=cell, groups=egroups,by_nuclide=True)
 List_Tallys_tmp[4] = mgxs.FissionXS(domain=cell,groups=egroups,by_nuclide=True)
 List_Tallys_tmp[5] = mgxs.NuFissionXS(domain=cell,groups=egroups,by_nuclide=True)
 List_Tallys_tmp[6] = mgxs.Chi(domain=cell,groups=egroups,by_nuclide=True)
 List_Tallys_tmp[7] = mgxs.ScatterMatrixXS(domain=cell, groups=egroups, by_nuclide=True)
 List_Tallys_tmp[7].correction = None
 List_Tallys_tmp[7].order = 1
 List_Tallys_tmp[8] = mgxs.NuScatterMatrixXS(domain=cell, groups=egroups, by_nuclide=True)
 List_Tallys_tmp[8].correction = None
 List_Tallys_tmp[8].order = 1

 List_Tallys+=[List_Tallys_tmp]
 name_region= cell.name
 List_Names_Tallys+=[["transport cross section - "+name_region,\
                      "total cross section - "+name_region,\
                      "capture cross section - "+name_region,\
                      "absorption cross section - "+name_region,\
                      "fission cross section - "+name_region,\
                      "nufission cross section - "+name_region,\
                      "chi - "+name_region,\
                      "scattering matrix - "+name_region,\
                      "n2n matrix - "+name_region]]
```

Figure 21. Part of the OpenMC Inputs Required for Cross Section Generation

The GenISOTXS code has been updated to read cross section data from the OpenMC cross section output. Figure 21 shows a keyword for each cross section type required in the OpenMC input in order to use GenISOXS as a post-processor. Using homogeneous and heterogeneous inputs with various compositions, it was confirmed that OpenMC generates a complete set of cross section data required for ISOTXS.

Cross sections generated from Monte Carlo codes work well for many cases; however, accuracy problems remain unresolved for higher-order scattering matrices and uncertainties in less dominant regions or isotopes, etc.

2.6 PERSENT

The PERSENT code was successfully modified to allow the computation of component uncertainties. The typical approach to calculate the uncertainty U is to use the covariance matrix C and computed uncertainties s.

$$U = \sqrt{s^T \cdot C \cdot s} \ , \tag{30}$$

The desired alteration was to take two sets of sensitivities s_1 and s_2 as input and compute

$$T = s_1{}^T \cdot C \cdot s_2 \ , \tag{31}$$

This particular computation is needed to compute the variance associated with a given uncertainty computation given multiple measurements. The source code change to PERSENT was very easy and completed in one day with the addition of one more verification problem for the nightly regression. The manual was also updated.

In addition to this minor change of PERSENT, the work to improve the performance of the sensitivity coefficient calculation was started. The sensitivity calculations in PERSENT are likely the slowest conceivable way to approach the problem whereas the approach taken for the perturbation problems is very efficient. The primary reason for the poor sensitivity performance is due to the initial trouble experienced trying to get the variational functional working which consumed all of the development time rather than spending some of it finding an efficient way of performing the operations. After some effort, a more robust way of applying the sensitivity coefficients has been devised but it will require several months to completely rebuild all of the sensitivity options of the PERSENT code.

2.7 BuildBot

Updates were made to the nightly regression test system of PROTEUS-SN, named BuildBot. First, the BuildBot tests were moved from the “Simon” machine to a newer machine called “Seville” with more recent operating system (CentOS Linux release 7.2.1511) which provided many compiler and tool upgrades. We note that test logs on Simon are being retained for the foreseeable future. The BuildBot configuration scripts and dependencies were installed on Seville as needed. Dependent libraries for PROTEUS-SN were also installed. The BuildBot server runs the nightly regression tests for PROTEUS-SN and the legacy tools (DIF3D, REBUS, etc.).

PROTEUS-SN is being tested for Intel 13.1 and GNU 4.8.5, and the Exodus Converter mesh converter tool is also being tested for these compilers. Notably, all the previous Intel 10.1 tests for the legacy tools were dropped from the nightly regression tests because this compiler is no longer available from Intel. Instead, the legacy tools are being tested with the newer Intel 12.1 and Intel 13.1 compilers. Also notably, support for compiling the legacy tools with the GNU compiler set had to be dropped completely as GNU no longer supports the Fortran 66 features prevalent in these codes. The new BuildBot Waterfall page which reports the test status is available from within the NE Division network.

3. Verification Tests

The improved capabilities of PROTEUS are verified using actual benchmark problems. The flexible thermal expansion capability is tested using ABTR with practical radial core expansion simulations. PROTEUS solutions are compared to the conventional approach with PERSENT. Experimental reactors with thermal spectrum including TREAT and RCF are simulated with PROTEUS. In this study, PROTEUS results are compared with Serpent Monte Carlo solutions, eventually targeting comparison with measurements for code validation in future.

3.1 Radial Core Expansion Simulations with PROTEUS-SN

Radial core expansion in sodium-cooled fast reactors provides an important reactivity feedback effect. As the reactor power increases due to normal start up conditions or accident scenarios, the core and surrounding materials gradually heat up causing both grid plate expansion and bowing of the assembly ducts. The core restraint system can be designed to guide the assembly movement into a free bow or limited free bow shape, both of which introduce negative reactivity feedback to the core. As time proceeds, the assembly deformation changes include both compaction and expansion at different regions, but once the assemblies contact each other at the above core load pad (ACLP) elevation which is part of the restraint system, the reactor is “locked up” and unable to compact any further [19], shown in Figure 22(c). Typically the core is designed such that this locked up state provides negative reactivity from the base configuration, and any further heating will cause outward bowing of the fuel assemblies and hence further negative reactivity feedback. Therefore, as the power and temperature increases, there exists a natural upper limit to these quantities as negative reactivity will be inserted once thermal expansion reaches the locked up state.

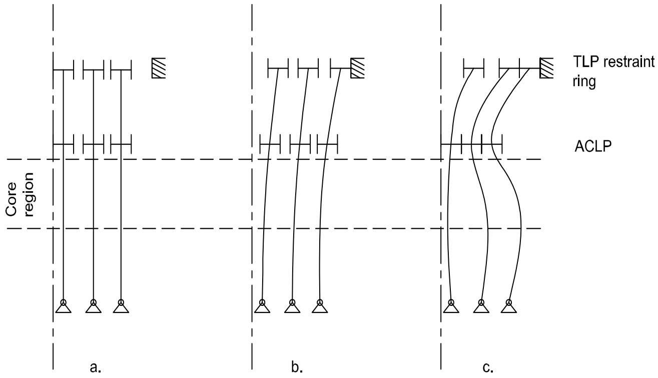


Figure 22. Deformation States for a Limited Free Bow Restraint System showing a) Undeformed (Base State), (b) Intermediate Deformation State, (c) Limited Free Bow (end state).

3.1.1 Conventional Modeling of Deformed Geometries

Given the importance of this negative reactivity feedback to accident scenarios, in particular, it is important to be able to estimate it accurately. The geometry limitations of conventional codes prevent the direct modeling of the radial core expansion. Instead, an indirect procedure [20] is used to estimate the reactivity feedback due to such an expansion:

1) Perform a DIF3D neutronics calculation of the undeformed (base) state to get the eigenvalue, forward flux, and adjoint flux.
2) Separately perturb the number densities of fuel, sodium, and structural isotopes in selected regions and use the perturbation and sensitivity analysis code VARI3D to calculate the reactivity coefficient due to these perturbations ($\frac{\partial \rho}{\partial D_{i,m}}$, where $\delta D_{i,m}$ is the change in density of material m in region i). The density perturbation is meant to emulate core expansion/contraction.
3) Use SuperEnergy-2 (temperature data) and NUBOW-3D (mechanical deformation code restricted to 1/12 core symmetry) to calculate the region-wise deformation and density changes due to thermal expansion. In particular, NUBOW-3D processes the VARI3D mesh-based reactivity coefficients to obtain the total core expansion reactivity coefficient for the particular deformation scenario.

The conventional procedure has some key assumptions and approximations: the density perturbations must be small (valid regime of perturbation theory), the effect of the geometrical expansion coupled with the density perturbations are neglected, the density perturbations of different isotope types (fuel, sodium, structure) are decoupled, the coupling effect of space-variation of density perturbations in different areas is ignored, and 1/12 core symmetry restriction is imposed by NUBOW-3D. It is therefore of interest with today's advancements in modeling and simulation to model the radial core expansion phenomena more directly, providing some confidence in the accuracy of the radial core expansion reactivity feedback coefficients used in design today (in some cases, these coefficients are not used at all due to lack of confidence in their accuracy).

3.1.2 PROTEUS-SN Advanced Modeling of Deformed Geometries

A new capability has been added to PROTEUS-SN in order to model the deformed configuration directly, including automatic processing of material densities to preserve mass and/or density of each isotope. Special handling of isotopes is done through "material models", for example, sodium can be backfilled into any expanded geometry which amounts to an increase in mass. PROTEUS-SN has always been able to perform deformed geometry configurations, but previously there was no way to obtain a deformed mesh except direct coupling to a structural mechanics code such as Diablo through SHARP. Additionally, there was no way to automatically handle material density changes except manual pre-processing techniques which are error prone and labor intensive.

We have removed these limitations by creating a simplified mesh deformation tool (see Section 2.4.2) that applies both axial and radial expansion of any 3D mesh according to user-specified strain functions in the X-, Y-, and Z-dimensions. This new mesh deformation tool applies simple deformations where all assemblies are assumed to shrink/expand together in the same shape. While in reality assemblies should be permitted to deform according to unique axial shapes (outer vs. inner, fuel vs. reflector, etc.), this tool provides an approximate deformation scheme for testing purposes.

Then, a capability to automatically process the deformed mesh as well as material density adjustments was implemented in PROTEUS-SN for standalone simulations. To run a deformed mesh simulation, the user needs only to provide the full inputs for the base (undeformed) configuration, as well as the deformed mesh, and the code processes everything else automatically. This eliminates the need for pre-processing isotopic densities to account for mesh deformation. Liquid sodium coolant can be backfilled (or removed) in any deformed region in the core using a tag called a Material Model in the material assignment file and a flag for PROTEUS-SN to compute the density based on material models. This is useful in cases where the sodium density, rather than mass, should stay constant in the deformed case.

3.1.3 Calculation of Radial Core Expansion Reactivity Feedback in the ABTR Design

To evaluate the new PROTEUS-SN capability, the legacy procedure was simplified to use only DIF3D-VARIANT and PERSENT (replacement for VARI3D), omitting the need for temperature and structural mechanical feedback data. In this new procedure, a deformation shape is assumed (not calculated), and then applied to PERSENT mesh-based reactivity feedback coefficients to compute the total reactivity feedback, as shown in Figure 23.

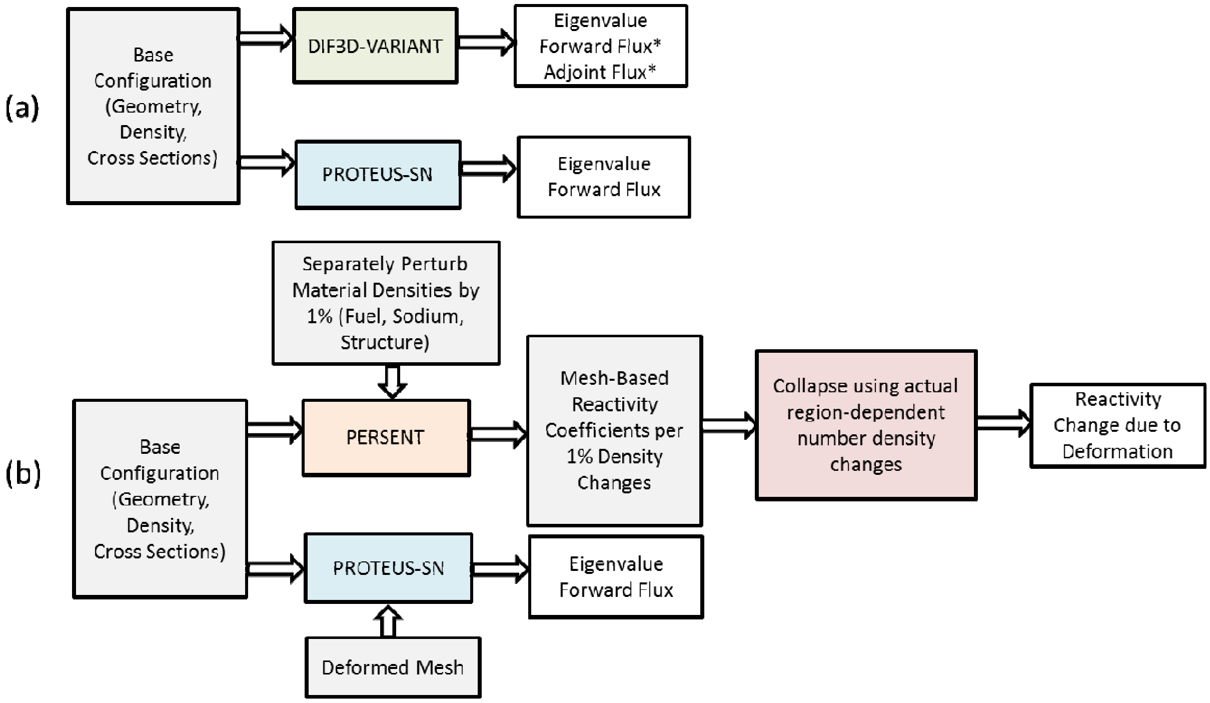

(a)
Base Configuration (Geometry, Density, Cross Sections)
DIF3D-VARIANT
Eigenvalue
Forward Flux*
Adjoint Flux*
PROTEUS-SN
Eigenvalue
Forward Flux
(b)
Separately Perturb Material Densities by 1% (Fuel, Sodium, Structure)
Base Configuration (Geometry, Density, Cross Sections)
PERSENT
Mesh-Based Reactivity Coefficients per 1% Density Changes
Collapse using actual region-dependent number density changes
Reactivity Change due to Deformation
PROTEUS-SN
Eigenvalue
Forward Flux
Deformed Mesh

Figure 23. Consistent Workflow to Compare Perturbation-based and Directly Calculated Reactivity Feedback from Radial Core Expansion: (a) Undeformed Mesh Simulation and (b) Deformed Mesh Simulation.

The PERSENT reactivity feedback coefficients are calculated using the base configuration DIF3D-VARIANT flux file along with material density perturbations. The assumed deformation shape is then modeled directly using PROTEUS-SN by processing the base configuration mesh with the mesh deformation utility tool described previously. The two neutronics models (DIF3D and PROTEUS-SN) are entirely consistent (same geometry, cross section data, and homogenized assembly regions).

The Advanced Burner Test Reactor (ABTR) core was chosen for this test as DIF3D inputs were available from past work. The material compositions and axial dimensions may differ slightly from previous PROTEUS-SN models of the ABTR. The ABTR core, shown in Figure 24, consists of 199 assemblies arranged in a hexagonal grid with 9 rings. The core contains three different fuel assembly types as well as control, shield, and reflector assemblies. The "Material Test" assemblies were replaced with reflector assemblies in the model. Axially, the core extends from 0.0 cm to 345.68 cm elevation where the active core is located between 110.54 cm and 194.95 cm elevation. The assembly pitch is 14.685 cm.

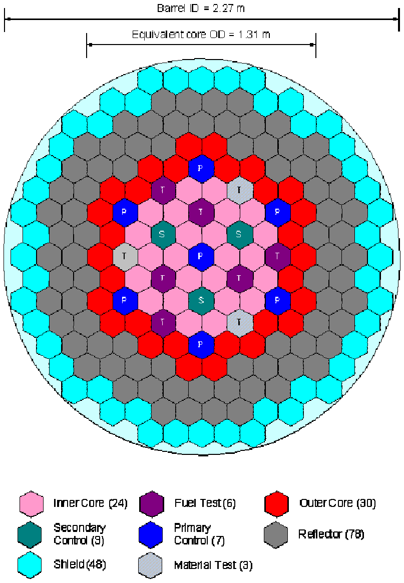


Figure 24. Core Map of the Advanced Burner Test Reactor.

The base configuration (undeformed geometry) with homogenized assemblies was simulated with PROTEUS-SN and DIF3D-VARIANT, and results are summarized in

Table **13**. Multigroup flux distributions and power distributions calculated in PROTEUS-SN are shown in Figure 25 and Figure 26. Note that power distributions are discontinuous across assembly boundaries due to material changes. A PN convergence study was partially performed for DIF3D-VARIANT which showed increasing eigenvalue as the PN order was increased, as predicted. Recent work [21] has shown that L7T7 cubature (SN method) is sufficiently converged for PROTEUS-SN for the ABTR homogeneous geometry configuration.

Table 13. Eigenvalues Computed for the Base (Undeformed) State of the ABTR.

Code and Options	Base (Undeformed) Eigenvalue
DIF3D-VARIANT 040601 space, 41 axial regions (8-10 cm mesh), P1 scattering	1.02087 (P3) 1.02226 (P5) 1.02273 (P7)
PROTEUS-SN Quadratic FEM (6 triangles/ assembly, 41 axial regions), L7T7 cubature (128 angles on sphere), P1 scattering	1.02119

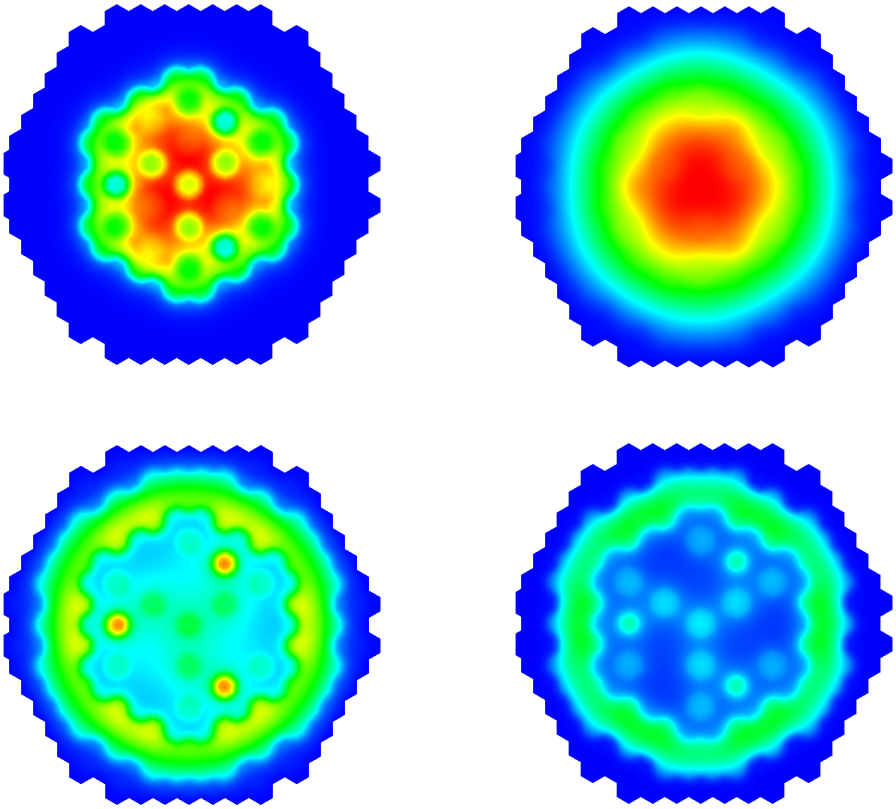


Figure 25. PROTEUS-SN Computed Multigroup Fluxes at Plane Z=150 for Base Configuration. Clockwise from upper left: Group 1 (6.06-14.19 MeV), Group 10 (67.4-111.1 keV), Group 17 (2.03-3.35 keV), Group 19 (0.454-1.23 keV).

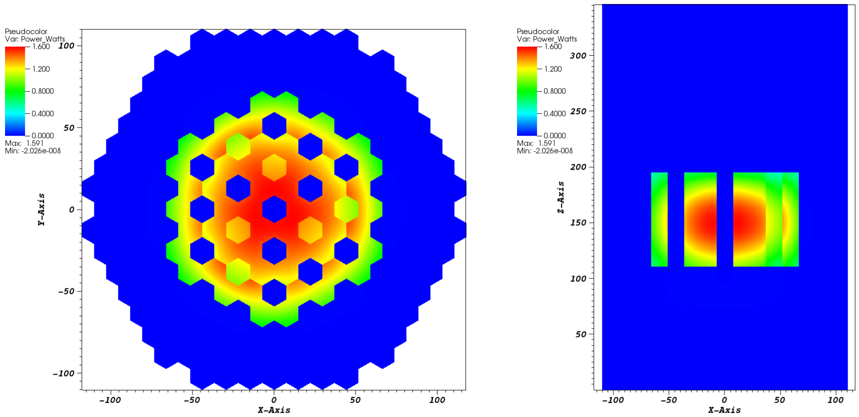


Figure 26. PROTEUS-SN Normalized Power Distribution at Z=150 (left) and Y=0 (right) for Base Configuration.

Mesh convergence studies were not performed for either code due to time limitations, but reasonable accuracy is expected using the parameters shown. We note that DIF3D and PROTEUS-SN solutions are expected to have 100-200 pcm discrepancies due to the inconsistent χ fission neutron distribution treatment in the two codes. The two codes are in agreement within this tolerance for the base configuration.

As illustrated in Figure 23(b), the DIF3D/PERSENT procedure to compute reactivity feedback from radial core expansion is a two-step procedure. In the first step, generalized 1% fuel, structure, and sodium density perturbations are applied to the base configuration to estimate region-dependent reactivity changes, $\frac{\partial \rho}{\partial D_{m,i}}$, where m is the one of {fuel, structure, sodium} and i is the mesh region. This calculation is not dependent on the actual deformation. These region-wise quantities are then post-processed in a second step using the region-wise material density changes in the deformed state, $\Delta D_{m,i}$, to yield the total reactivity change:

$$\Delta\rho = \sum_{m=fuel,sod,struct} \sum_{i=1}^{N} \left(\frac{\partial \rho}{\partial D_{m,i}} \right) \Delta D_{m,i} \tag{32}$$

The whole core reactivity feedback coefficients, i.e., the reactivity introduced for a 1% change in density of material {fuel, sodium, and structure} over the entire system are also calculated by PERSENT for the ABTR core and reported in Table 14, although these quantities are not directly used in these calculations. These coefficients were computed for both P3 and P5 angular order in VARIANT, and observed to be nearly independent of the PN order. Note that the sodium density reactivity coefficient fluctuated from negative to positive upon increasing the PN order, but this coefficient is so small compared to the fuel and structure coefficients, it can be neglected. Additionally, in this radial core expansion study, we maintain constant sodium density, so this factor is not used.

The region-wise reactivity coefficients were calculated on a mesh consisting of 41 8-10 cm axial segments (11 segments within the active core zone) and 9 radial rings. Similar assemblies within the same ring were grouped together to take advantage of radial symmetry. Since the perturbations applied in this work are all symmetric about the core center, all assemblies within a given ring experience the same density changes and therefore this grouping should not matter. Slices along Y=0 (centerline along the Y-axis) were taken of the region-wise reactivity coefficients computed by PERSENT and plotted in Figure 27 (fuel perturbation) and Figure 28 (structure and sodium perturbations).

Table 14. Whole Core Reactivity Coefficients for the ABTR Reactor Calculated by DIF3D-VARIANT/PERSENT per 1% Increase in Fuel, Structure, and Sodium Density.

	PERSENT + DIF3D-VARIANT P3	PERSENT + DIF3D-VARIANT P5
$\frac{\partial \rho}{\partial D_{fuel}}$	4.53901E-03 (+454 pcm)	4.52323E-03 (+452 pcm)
$\frac{\partial \rho}{\partial D_{structure}}$	2.09060E-04 (+21 pcm)	2.02391E-04 (+20 pcm)
$\frac{\partial \rho}{\partial D_{sodium}}$	6.91301E-06 (+0.7 pcm)	-3.66336E-06 (-0.4 pcm)

Figure 27 shows that the fuel assemblies closest to the core center have the highest reactivity worth, as expected. Additionally, the axial middle of the core has the highest reactivity worth. The reactivity worth of a fuel assembly is at most 15 pcm (1.5E-6 on the chart) per 1% change in fuel density, per axial zone (8-10 cm segment). The fuel assemblies have strictly positive reactivity worths because an increase in fuel density anywhere in the core implies core compaction or an increase in fissionable material and production of neutrons.

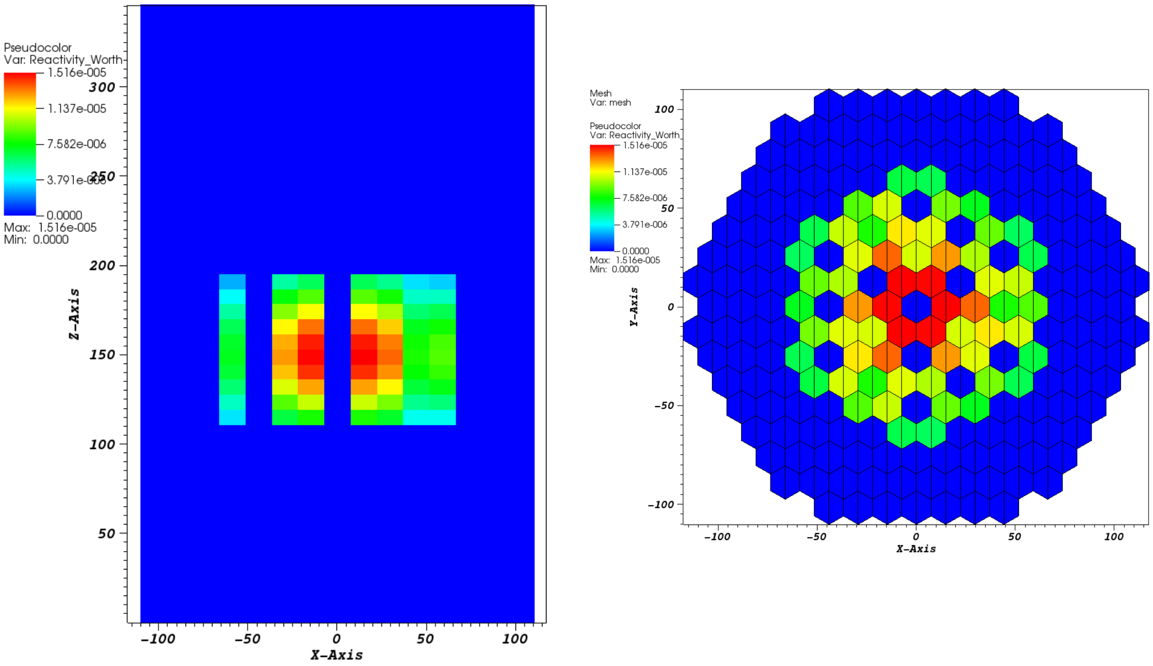


Figure 27. Region-Dependent Reactivity Worth Calculated by PERSENT (P5) for 1% Increase in Fuel Density. (left) Slice taken at Y=0, (right) Slice taken at Z=150 cm (active core zone).

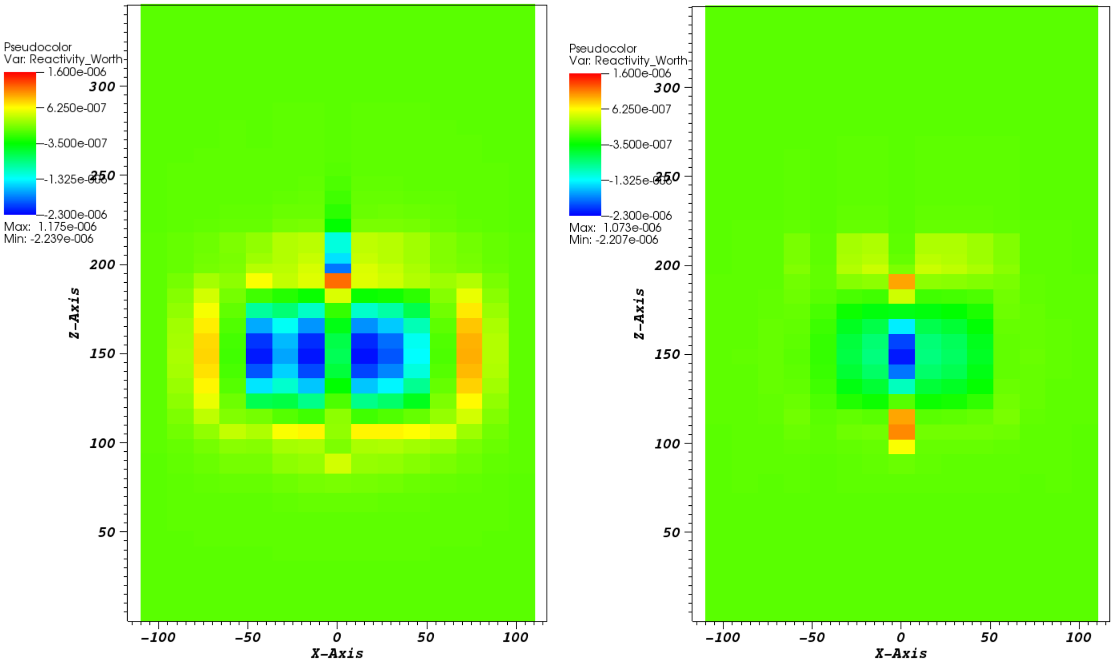


Figure 28. Region Dependent Reactivity Worth Calculated by PERSENT (P5) for 1% Increase in Structure Density (left) or Sodium Density (right). Both slices taken at Y=0.

Figure 28 shows the reactivity worth for 1% perturbation in structural density (left) or sodium density (right). The legend extents on these two plots have the same color scale for direct comparison. One can immediately note that the maximum region-wise reactivity worths of both structural and sodium density changes are roughly 5-10x smaller in magnitude than those due to fuel density changes. Thus the reactivity feedback will be dominated by fuel density changes. Also notable is that sodium density reactivity worth is negligible (<<1 pcm) in nearly all regions except the central control rod where it is negative (-2.3 pcm) in the center of the active core and positive at the bottom and top of the active core. Overall the mesh-dependent sodium reactivity worths appear to roughly cancel each other out provided a uniform core-wide perturbation is applied. This is consistent with the very small (~-0.4 to 0.7 pcm) values reported in Table 14.

Looking at these values, we should expect the perturbations performed at the axial center of the active core to have the largest impact on reactivity. Perturbations well above and below the active core should not result in large reactivity changes.

A variety of deformation cases were strategically chosen to compare accuracy of the legacy technique to a direct modeling approach (PROTEUS-SN). The deformations are applied such that the pitch of all assemblies in an X-Y plane (same Z-elevation) changes identically. The change in assembly pitch is plotted as a function of Z for the different cases in Figure 29 and tabulated in Table 15. Data points are linearly interpolated between specified Z-elevations.

Case I is a free flowering shape based on the assumption that fuel assemblies should bow no more 0.2 mm from centerline. Grid plate expansion is ignored, and the deformations are assumed to represent bowing. The general shape chosen is similar to that in Figure 22(b) with representative numbers chosen after discussion with an expert in nuclear reactor structural mechanics for a different reactor design [22].

Case II is representative of an intermediate state in a limited free bow design (not locked up yet), whereby the expansion in the active core initially bows inward, producing positive reactivity. More information on this is given in Reference [19].

Case III applies a uniform radial grid expansion of 0.596% (which yields a change in assembly pitch of +0.88 mm based on a 146.85 mm base pitch). This radial expansion factor at operating condition was taken from the ABTR Specification Report [23].

Case IV and Case IV incorporate the uniform radial grid expansion from Case III as well as the deformation shapes from Case I and Case II, respectively.

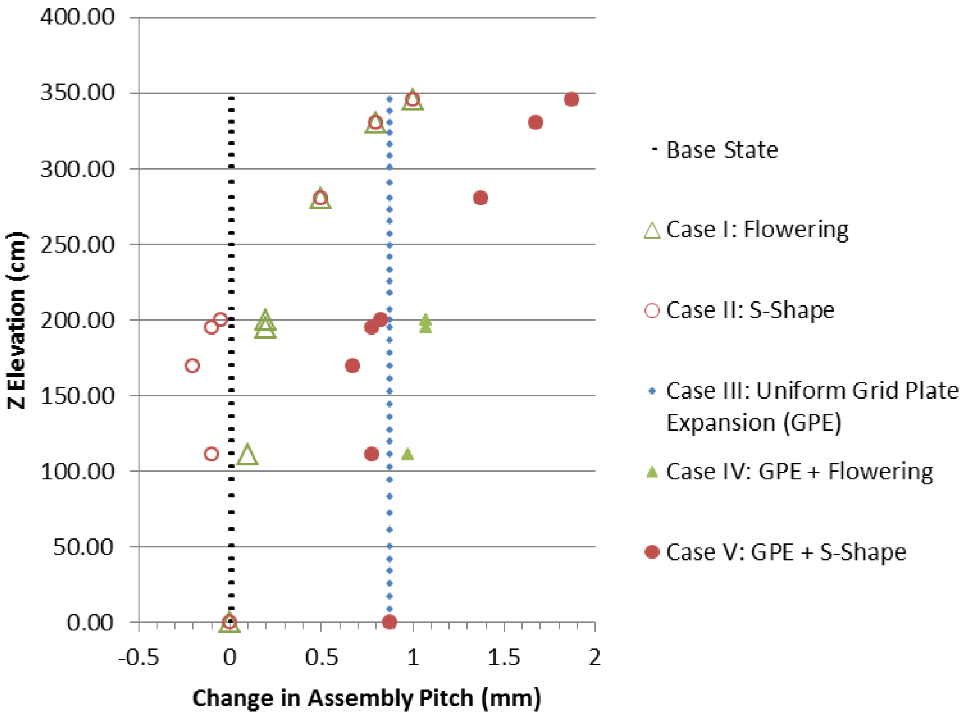


Figure 29. Radial Core Deformation Schemes Applied for ABTR Analysis.

Table 15. Radial Core Deformation Schemes Applied for ABTR Analysis.

		Change in Assembly Pitch (mm) Base Case Assembly Pitch=146.85 mm				
Geometry	**Elevation (cm)**	**Case I**	**Case II**	**Case III**	**Case IV**	**Case V**
Above Active Core	345.68	1.0	1.0	0.88	1.88	1.88
	330.61	0.8	0.8	*	1.68	1.68
	280.36	0.5	0.5	*	1.38	1.38
ACLP Elevation	200.0	0.2	-0.05	*	1.08	0.83
Active Core	194.95	0.2	-0.1	*	1.08	0.77
	169.63	*	-0.2	*	*	0.67
	110.54	0.1	-0.1	*	0.98	0.78
Bottom	0.0	0.0	0.0	0.88	0.88	0.88

We emphasize that the deformation shapes modeled applied here are contrived and at best considered to be somewhat “representative” - they are not implied to be the true deformation that would occur. To obtain the true deformation, direct coupling with structural mechanics code and complete description of the core restraint system would be required via the SHARP Toolkit (in fact, this was previously done in past years under NEAMS but without the automatic density adjustment capabilities new to this year’s work). The purpose of this work is to compare conventional techniques to the new direct modeling approach using consistent models and deformations for fair comparison. Additionally, contrary to real deformation shapes, typically the outer reflector and shield assemblies do not bow and deform as much as indicated here.

PROTEUS-SN directly computes the volume change in each mesh region (with regions identically defined in DIF3D), so the volume changes due to deformation are easily translated to isotopic density changes and processed against the PERSENT region-based reactivity coefficients. Direct simulations of the deformed mesh were performed using PROTEUS-SN. The eigenvalue and reactivity worth results are summarized in Table 16. As expected, all of the reactivity worths are negative expect for Case II, since Case II is the only deformation scheme that pushes fuel material closer to the core centerline than the based configuration. While flux and power solutions were easily generated for the deformed cases, it is not worthwhile to compare them visually against the undeformed base case, since differences are too small to discern and the powers are normalized inside the code before printing.

The PROTEUS-SN-computed reactivity worths for Case I (-70) and Case II (+69) are nearly identical, just opposite in sign, even though the deformation shapes are significantly different. However the deformation shapes within the active core are similar (but in opposite directions). The region-wise distribution of reactivity worths computed by PERSENT predicted that only changes near the active core would contribute non-negligibly to reactivity worth, and this is confirmed in the comparison between Case I and Case II.

The PROTEUS-SN-computed reactivity worth of Case III, uniform grid plate expansion (-413 pcm) is much higher in magnitude than the initial Case I and Case II assembly bowing deformation schemes (-70 and +69 pcm respectively). Therefore the grid plate expansion is a significant effect that must be considered. Case IV is simply a superposition of the Case I and Case III deformation shapes and interestingly, the reactivity worth of this case is also the superposition of the reactivity worths of those two calculations. A similar statement holds for Case V. This suggests that these perturbations cause highly linear changes in reactivity, and the reactivity coefficients predicted by PERSENT/DIF3D-VARIANT are actually quite accurate for a large regime of values.

However, in all cases, the DIF3D-based PERSENT calculations overestimate the reactivity worth of a given deformation scheme by approximately 35% compared to PROTEUS-SN (direct computation). This overestimation is important to note, because it means that the legacy calculation is not conservative.

Table 16. Computed Reactivity Worths for ABTR Deformation Schemes.

		Predicted Reactivity Worth of Deformation				
Code	**Value**	**Case I**	**Case II**	**Case III**	**Case IV**	**Case V**
PERSENT/DIF3D-VARIANT (P5)	$\Delta\rho$ (pcm)	-95	+98	-558	-652	-463
PROTEUS-SN	k	1.02046	1.02191	1.01690	1.01618	1.01762
	$\Delta\rho$ (pcm)	-70	+69	-413	-483	-343
% Difference in Reactivity Worth	$\frac{PERSENT - PROTEUS}{PROTEUS}$	+35%	+42%	+35%	+35%	+35%

In reactor safety analysis, safety margins must be used to protect against uncertainty or error in calculations. While the legacy procedure produces answers that are comparable to PROTEUS-SN results and follow the same trends, the bias in the answer raises uncertainty in using this technique for arbitrary calculations. This bias is likely arising from the fact that geometry changes are not accounted for in the perturbation theory method, and therefore a 1% increase in fuel density perturbation is actually adding fuel to the system, when in reality the mass of fuel should be conserved. Additionally, the legacy procedure does not take into account combined effects, i.e. in these deformation schemes, the structure and fuel densities change together. The lack of modeling of the physics interplay between these two materials could be contributing to some of the error.

In conclusion, the legacy method produced reactivity worth values for radial core expansion of the ABTR which follow the expected trends, but are non-conservative by at least 35%. Utilizing the legacy-computed values for reactor design could result in overprediction of negative reactivity feedback for a radial core expansion scenario, possibly resulting in an unsafe operating regime.

We also note that the PROTEUS-SN calculations were consistently performed for homogeneous assembly geometry in this case. There is likely a bias effect due to non-explicit modeling of the assembly ducts and fuel pins. This should be explored in the future. Additionally, such heterogeneous modeling will be essential to deforming the mesh according to more complex functions of space.

In the future, we plan to add a capability to deform meshes based on NUBOW-3D data. That case requires more complex movement of assemblies and identification of assembly features which is not currently necessary or implemented in PROTEUS-SN since it is based on the general finite element method.

3.2 Thermal Reactors

3.2.1 TREAT

TREAT is a heterogeneous, air-cooled, graphite-moderated and graphite-reflected thermal reactor. A detailed description of TREAT can be found in the facility's 1960 design summary report **Error! Reference source not found.**]. The reactor is fueled with highly-enriched (~93%) UO_2 dispersed in graphite, with a fuel carbon-to-uranium (C/U) atomic ratio of roughly 10000 to 1. The fuel is arranged in zircaloy-clad fuel assemblies, with an approximately 4 ft long central fuel section and 2 ft long aluminum-clad graphite reflectors above and below the fuel. The assemblies are approximately 4 in x 4 in square, with chamfered corners.

The reactor core can accommodate a maximum of 361 assemblies, arranged in a 19x19 array. The core is surrounded by a permanent graphite reflector, which is in turn enclosed in a concrete bioshield. The reflector includes a set of movable blocks of graphite, which could be removed to provide viewing slots to the core. [24]

Modeling and simulation of TREAT is challenging due to a number of issues including long-standing uncertainties in core properties, gaps in TREAT knowledge created by the multi-decade standby in reactor operations, and limitations in the currently-available historic measured data. One of the most significant uncertainties is the boron impurity of the TREAT fuel. ANL is participating in the IRP NEUP project jointly led by Oregon State University (OSU), University of Michigan (UM), and Massachusetts Institute of Technology (MIT), in which two core configurations of TREAT, Minimum Critical Core (MinCC) and M8CAL as shown in Figure 30, are proposed as new benchmark problems of the IRPhEP handbook.

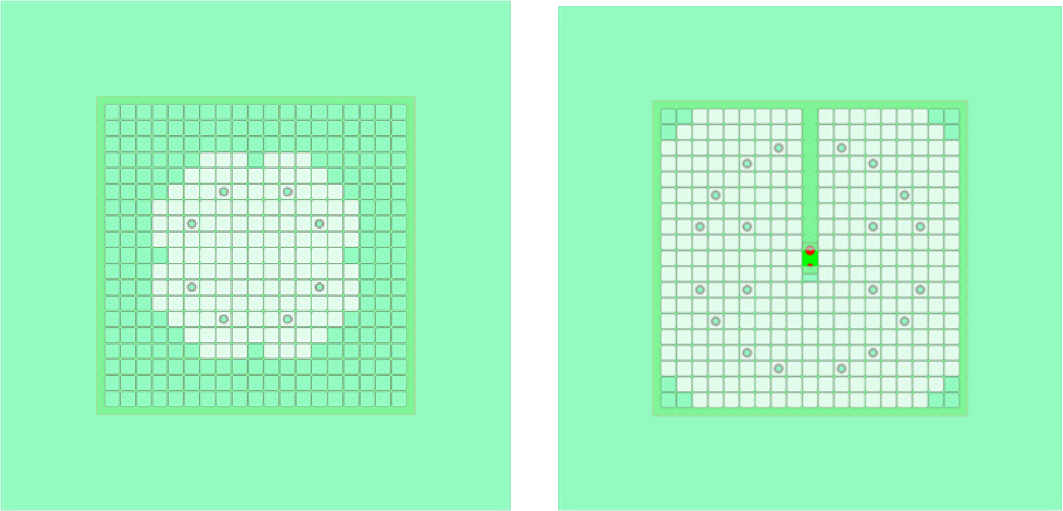


Figure 30. 2D View of MinCC (left) and M8CAL Core (right) of TREAT.

The conditions for MinCC and M8CAL finally determined by the IRP NEUP team are mainly 7.53 ppm boron and 267 ppm iron impurity in the graphite fuel, 59% graphitization, 16 Zr assemblies, and heated condition at 600 K. Core eigenvalues resulted from Serpent with the ENDF/B-VII.1 MCNP library are 1.00413 ± 0.00020 and 1.00394 ± 0.00020 for MinCC and M8CAL, respectively.

The MinCC and M8CAL experiments were used for verification and validation of PROTEUS. Due to air cooling channels, we use the MOCEX solver of PROTEUS, which is based on a rigorous formulation using 2D MOC radially and the Galerkin finite element method axially and is thus able to accurately solve a 3D problem with optically thin channels through which the axial neutron streaming has a significant impact on solutions. The calculation models for PROTEUS were constructed based on the two Serpent models shown in Figure 30. The geometries and meshes for components such as fuel, control rod, zirconium, and graphite blocks were generated by first using CUBIT, which were used to construct the 2D mesh since MOCEX needs a 2D mesh only.

The meshes used for analysis are shown in Figure 31. The “normal mesh” was initially generated for verification tests of PROTEUS with axially simple geometry cases, but later on the 2D projected meshes were generated to simulate actual 3D cores. The 2D projected mesh simultaneously includes features to represent geometries at different axial levels in the core. For example, the normal mesh type includes only 4 regions for the fuel block and only 9 regions for the control rod block, while the 2D projected mesh type includes 8 regions for the fuel block and 17 regions for the control rod block. For a 3D whole core calculation which needs large computation time and resources, the coarse 2D projected mesh is more practical than the fine one. Once the meshes for block components are created using CUBIT, 2D or 3D meshes were easily be merged by the mesh generation tool, MeshToMesh. Figure 32 and Figure 33 show the 2D core meshes for MinCC and M8CAL, respectively, constructed by MeshToMesh.

Macroscopic cross sections with 11 energy groups were generated using the Serpent models of MinCC with the ENDF/B-VII.1 library. Using the GenISOTXS code, macroscopic group cross sections generated from Serpent were converted to the ISOTXS format which PROTEUS can read. The lethargies of the 11 group structure are depicted in Figure 34. Rigorously, unique cross sections should be generated for each region with different material, location, and number density due to resonance self-shielding and neutron spectrum effects on the multigroup collapsing procedure. However, for this work, cross sections for similar materials at different locations were tallied together to reduce the number of cross section sets, which is expected to have only a minor impact on solutions.

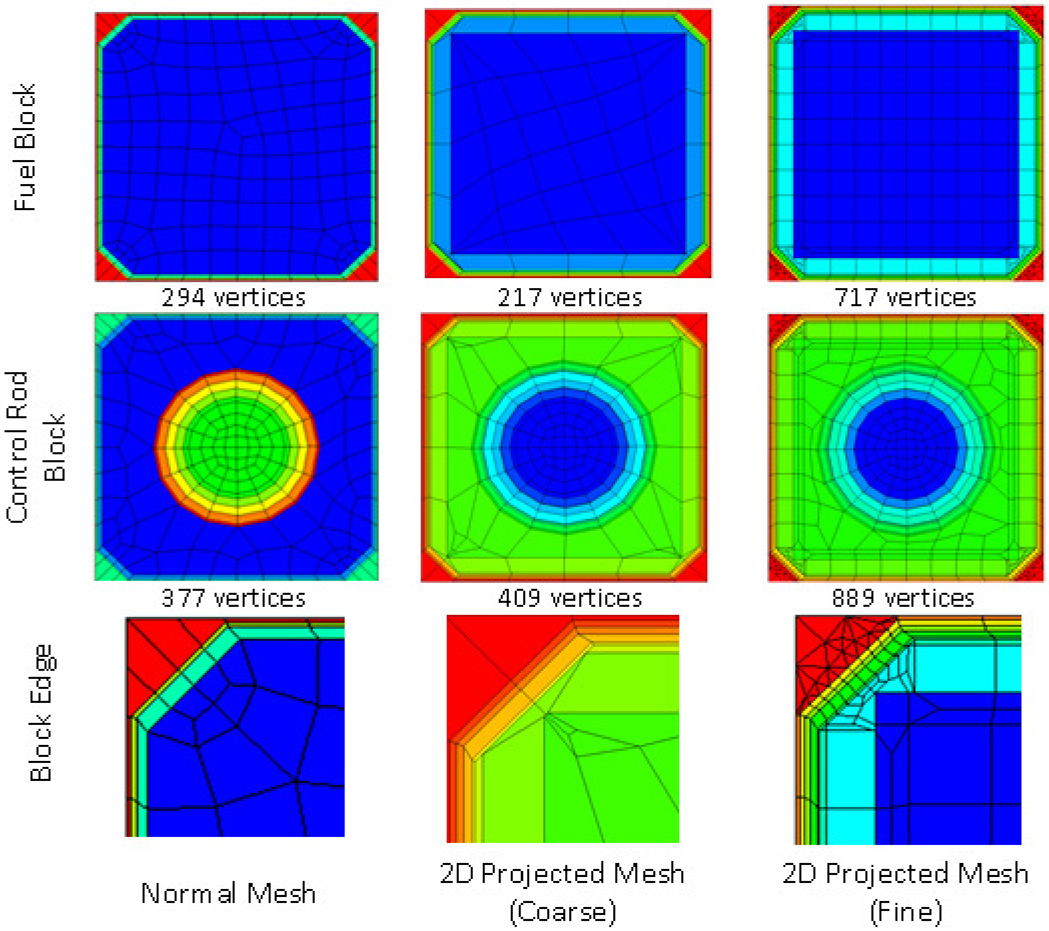


Figure 31. Meshes for Fuel and Control Rod Blocks.

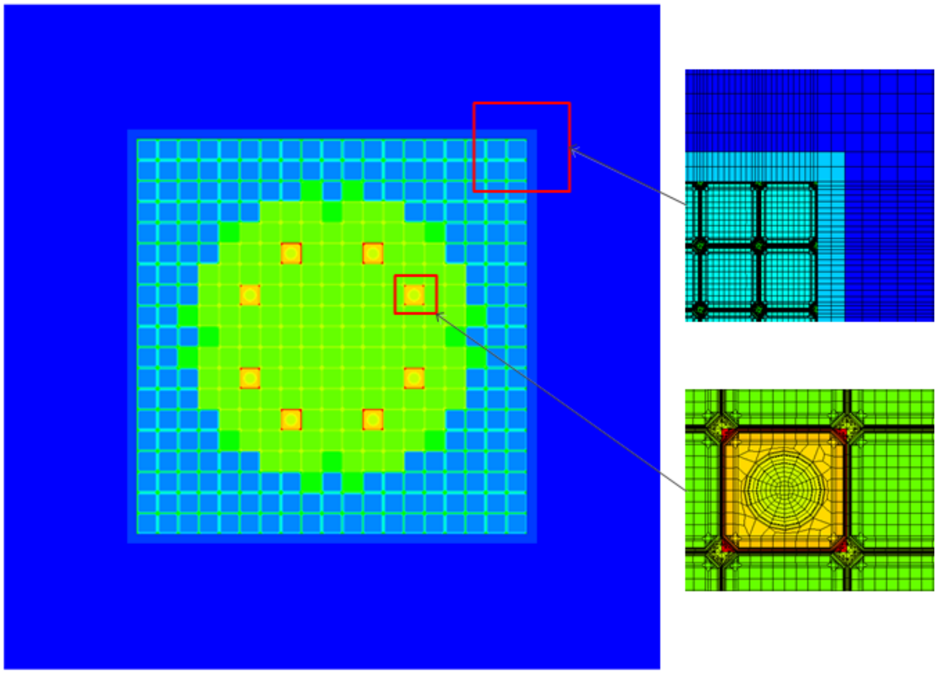


Figure 32. PROTEUS Model for the MinCC.

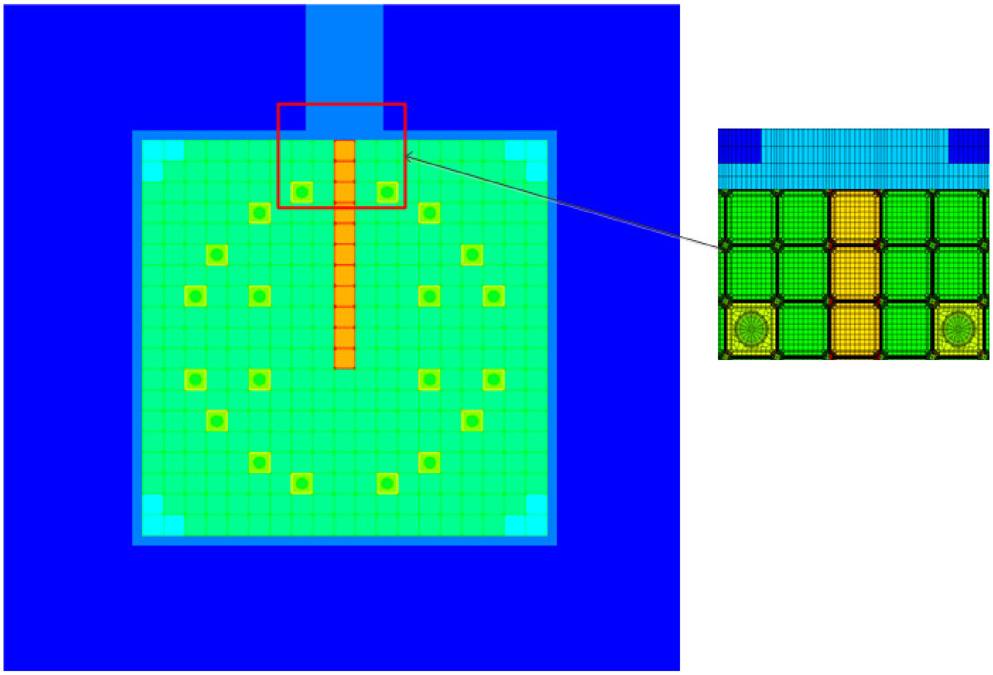


Figure 33. PROTEUS Model for the M8CAL.

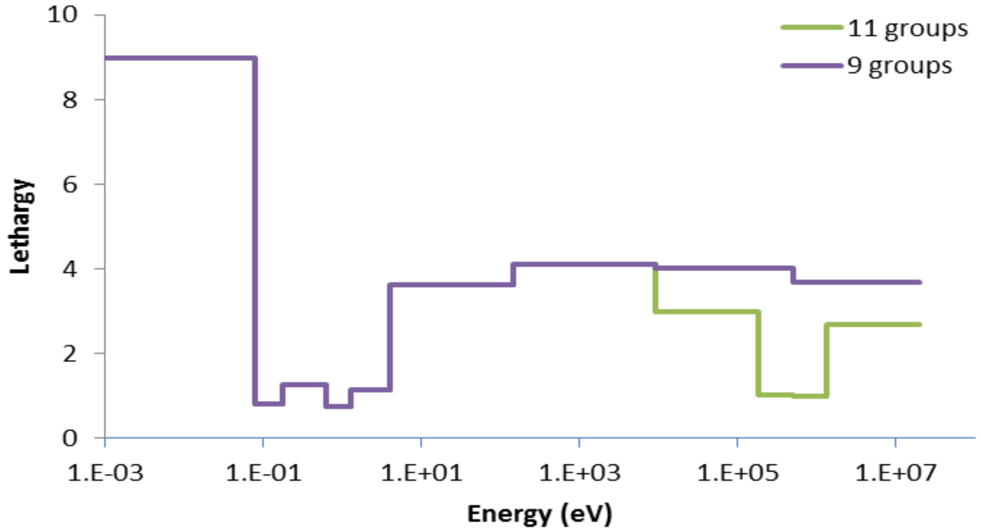


Figure 34. Group Structures Used in PROTEUS for TREAT Simulation.

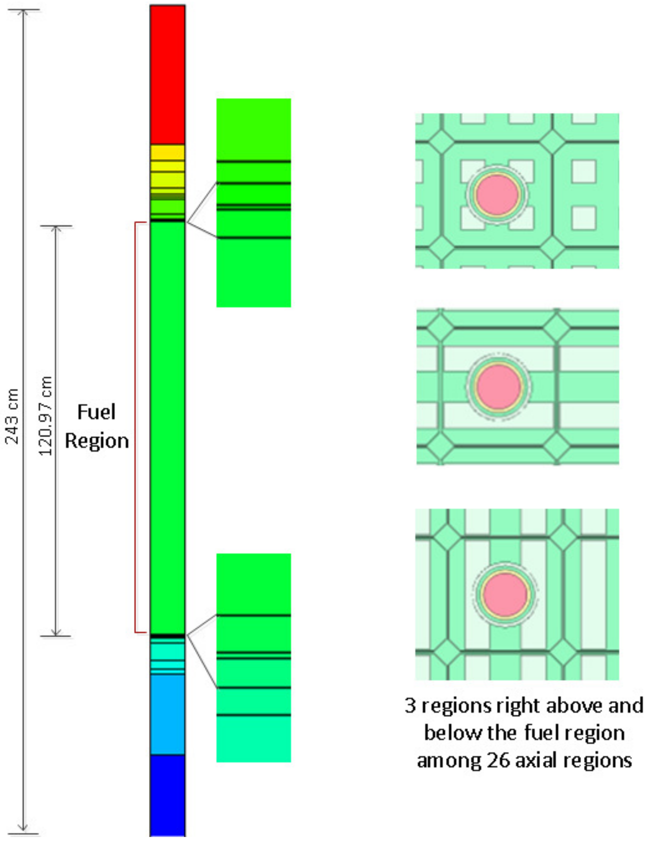


Figure 35. 26 Axial Planes and Geometry for MinCC of TREAT

For systematic verification tests, PROTEUS calculations were performed for single fuel block, 2D core with and without permanent graphite reflector (PGR), 3D single fuel block, and 3D partial core. An attempt was made to simulate 3D full core calculations, but it has not yet been completed because of large computation time and requirements. It should be noted that modeling the 2D projected mesh for the 3D whole core was challenging due to the axial geometric complexity shown in Figure 35.

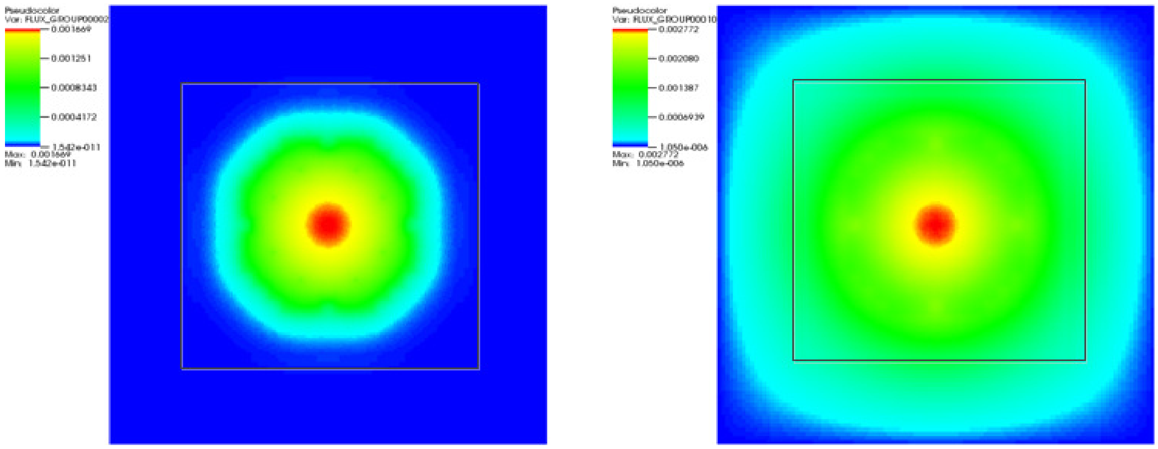
As shown in Table 17, PROTEUS eigenvalues for all cases agreed well with Serpent Monte Carlo solutions. In the case of 3D single fuel assembly (FA) case, the neutron streaming effect through the air cooling channel is up to 700 pcm. The PROTEUS eigenvalue is in good agreement with the Serpent solution, within 174 pcm. The eigenvalue difference between PROTEUS and Serpent tends to be reduced as a higher order of angular cubature from L5T15 to L15T15 in the polar direction is used. In addition, the 3D partial core case with a larger air channel representing the hodoscope based on M8CAL showed good agreement in eigenvalue between PROTEUS and Serpent solutions.

Figure 36 illustrates thermal and fast flux solutions of PROTEUS for the 2D core with permanent graphite reflector and without control rods based on the MinCC configuration. As expected, the flux level at the center of the core is the highest. Figure 37 shows 2D and 3D flux distributions of PROTEUS for the 3D partial core of M8CAL. The neutron streaming through the large air channel can be observed in the figure. For this case, reflective boundary conditions were applied radially, vacuum boundary conditions were used axially, and an angular cubature of L5T15 was used.

Table 17. Eigenvalues of TREAT Cases from Serpent and PROTEUS

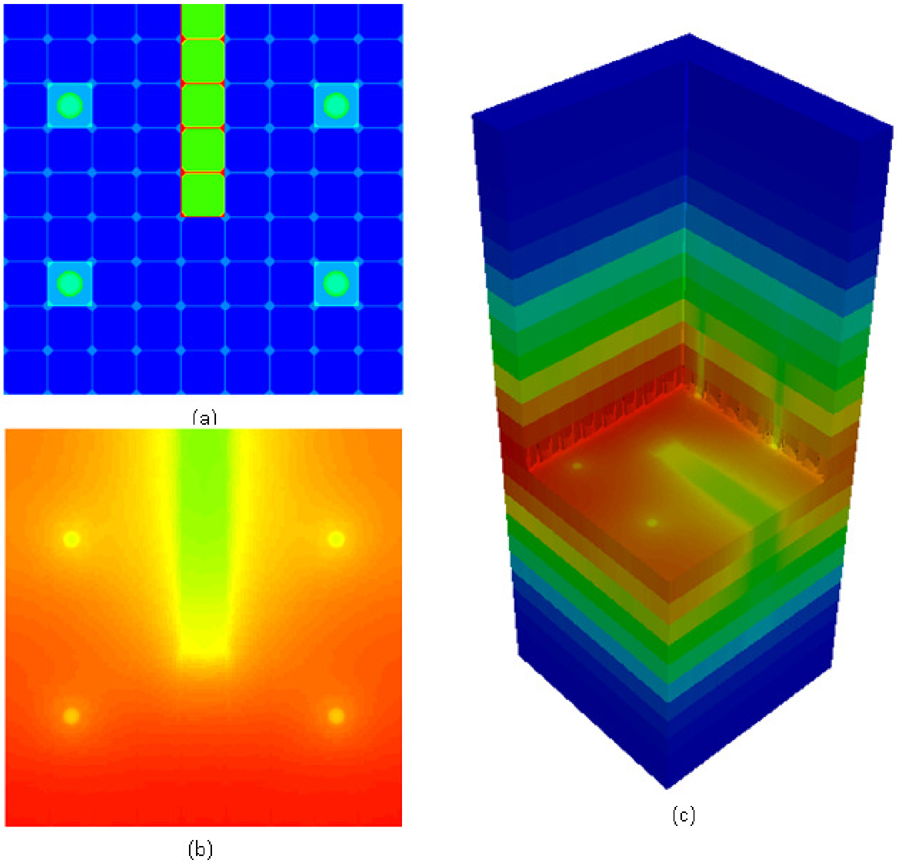
Case		Serpent	PROTEUS	
			Angle	Δk (pcm)
Fuel Block		1.66673 ± 0.00003	L5T15	-32
MinCC	2D Core w/o PGR	1.29939 ± 0.00015	L5T15	-167
	2D Core w/ PGR	1.23041 ± 0.00022	L5T15	337
	3D Single FA	1.45473 ± 0.00020	L5T15	220
			L9T15	196
			L15T15	174
M8CAL	3D Partial Core	1.37609 ± 0.00016	L5T15	264

* PGR: Permanent graphite reflector



(Fast Flux) (Thermal Flux)

Figure 36. PROTEUS Flux Solutions for 2D Core of MinCC.



(a) Loading, (b) 2D Fast Flux, (c) 3D Fast Flux

Figure 37. PROTEUS Flux Solutions for 3D Partial Core of M8CAL.

3.2.2 Reactor Critical Facility (RCF)

The RCF [9] is a heterogeneous, light water moderated thermal research reactor at RPI. With a maximum design power of only 100 watts, the RCF does not require a cooling system. The reactor is fueled with 357 low-enriched (~4.8%) UO_2 fuel pins arranged in 21x21 lattice configuration. The active fuel has an approximate height of 2.5 ft and an approximate width of 1.12 ft. Four square-type B4C control rods surround the fuel assembly lattice and consist of a stainless steel guide tube, a hydraulic brake and two neutron-absorber sections.

The NEUP project led by RPI and initiated in FY16, entitled "Development of Critical Experiments to Provide Validation Data for Multiphysics Coupling Methods," plans to perform multiphysics coupled simulations with PROTEUS-SN and Nek5000 of RCF experiments. To support the NEUP project and users, preliminary calculations of the RCF core were initially performed using PROTEUS-SN.

Compared to PROTEUS-SN, PROTEUS-MOCEX is more efficient for simulating thermal reactors with heterogeneous geometry and large flux gradients in the fuel and around neutron absorbers. However, PROTEUS-SN can also solve this type of problem if sufficient mesh refinement is provided to accurately represent flux gradients. Cross sections can be generated on the fly for thermal reactor problems using the Cross Section API (CSAPI) of PROTEUS which increases the computational time significantly due to the additional one-group fixed source problems. In this study, we used the Serpent Monte Carlo code rather than the CSAPI for cross section generation to reduce verification efforts as well as computation time.

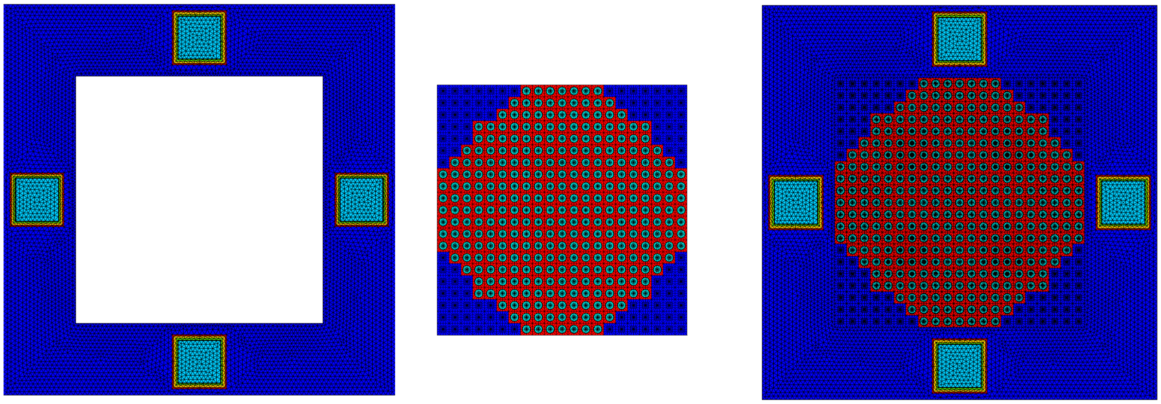


Figure 38. CUBIT mesh (left), Assembly UFmesh (center), Partial Core Mesh (right) of RCF**.**

For systematic verification tests, we constructed a single pin, 2D partial and whole cores, and 3D partial and whole cores. The mesh generation tool with UFmesh was used to generate a fuel pin mesh. It is trivial to construct meshes for a 2D partial core model (21×21 lattice) as well. Since the current version of UFmesh handles only Cartesian and hexagonal lattices, meshes for the regions outside the core were generated using CUBIT and then merged with the core mesh generated by UFmesh. The two mesh files (core and outside-core meshes) as well as the final merged mesh are shown in Figure 38. Care was taken to ensure that the two meshes were conformal on their merged boundaries.

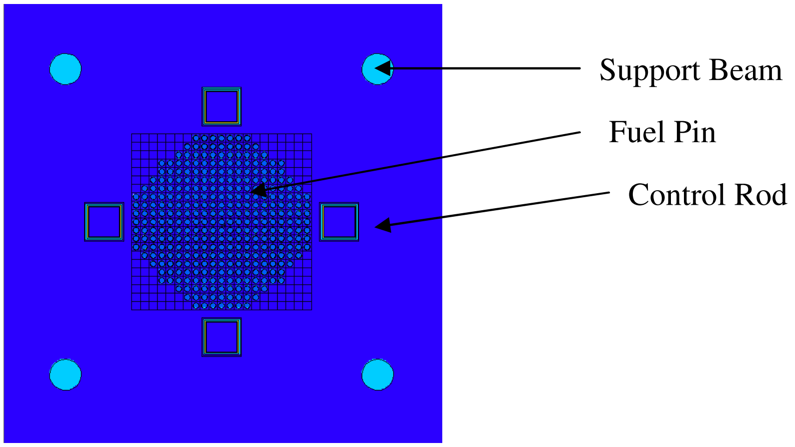


Figure 39. Top View of RCF Core.

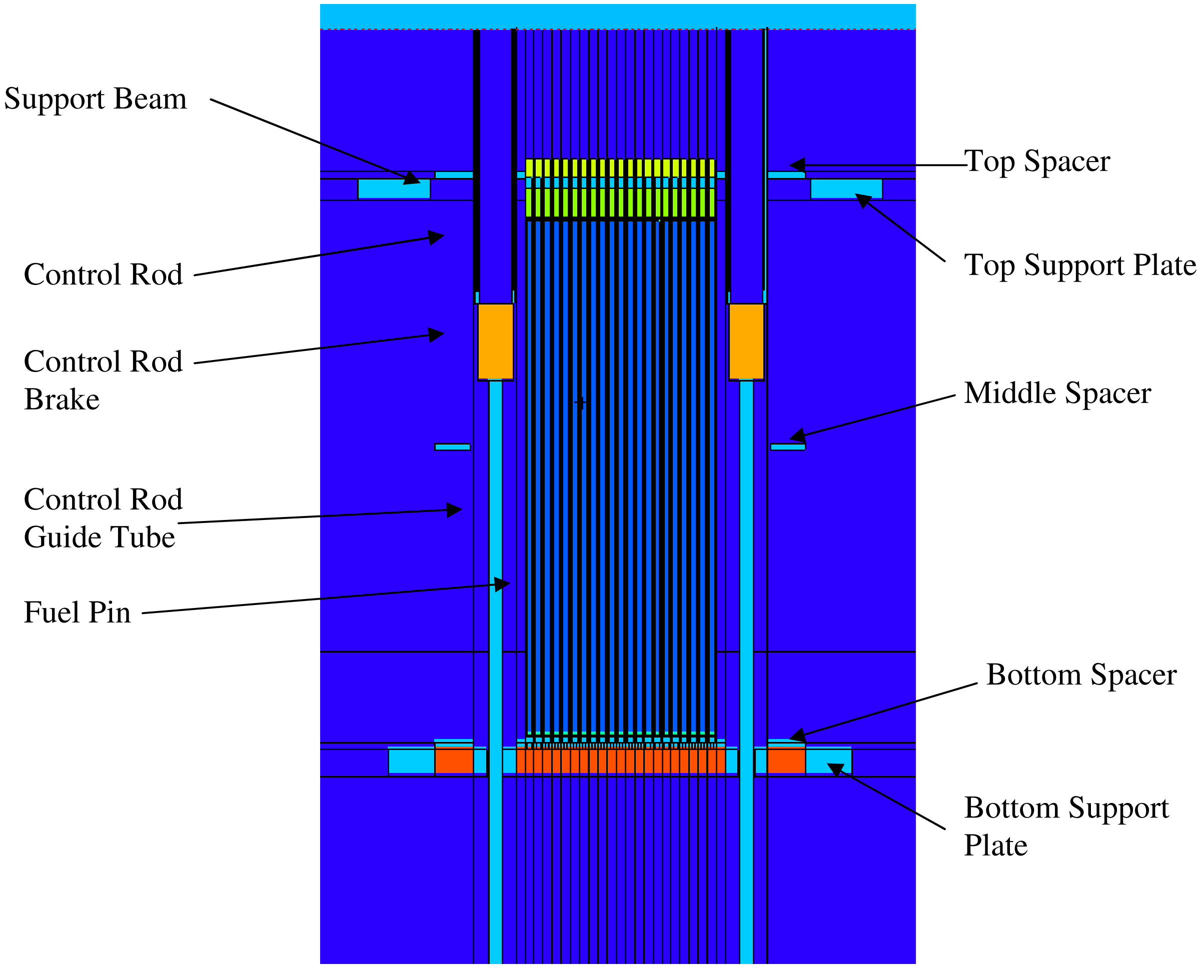


Figure 40. Side View of RCF Core.

PROTEUS calculations were performed for a single pin, 2D partial core and 2D whole core problems. 9-group or 11-group cross sections were generated from Serpent and processed into the ISOTXS format using the GenISOTXS code developed by ANL. The 9- and 11-group structures are compared in Figure 34, in which the 11 group structure includes more energy groups in the high energy range. Initially we used the 9-group cross sections but it was found that the 11-group cross sections produced more accurate solutions than the 9-group ones. For the pin cell problem, a Legendre-Tchebychev cubature order of L5T21, a scattering order of P3, and 577 vertex mesh were used, and for the 2D partial core problem, L3T3, P3, and total 26737 vertex mesh were utilized. Note that additional mesh and angular refinement studies should be performed for each problem to ensure the solution convergence. Eigenvalue solutions were compared with Serpent.

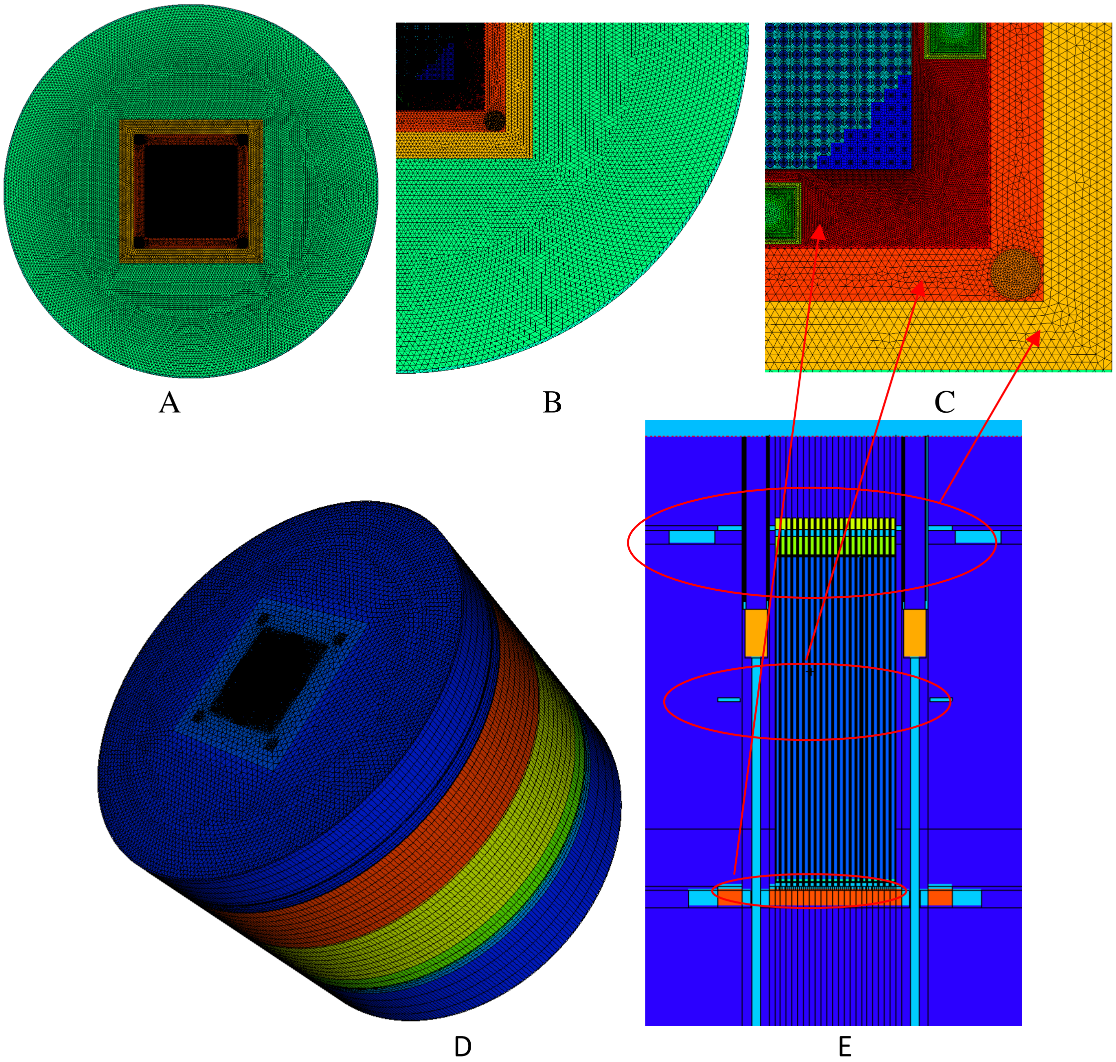
As shown in Table 18, PROTEUS eigenvalues with either 9-group or 11-group cross sections agreed reasonably well with Serpent solutions for the single pin and 2D partial core A models (Figure 38). For the 2D partial core B (Figure 39) and full core cases which include larger water reflector regions, however, PROTEUS with 11-group cross sections only produced eigenvalue solutions in good agreement with Serpent solutions. Control rod worths were in very good agreement between PROTEUS and Serpent. Figure 42 shows fast and thermal flux distributions produced from PROTEUS for a 2D partial core A (fuel region only) described in Figure 38. The flux solutions for 2D full cores with control rods in are illustrated in Figure 43.

Since the 11 group structure resulted in more accurate solutions than the 9 group structure based on other cases, it is expected that the PROTEUS solutions for the 2D full core problems would be significantly improved using the 11 group cross sections, following the trend in the 2D partial core B problem.

Table 18. Comparison of Eigenvalue between PROTEUS and Serpent

Model		Serpent	PROTEUS*		
Single Pin		1.41782 ±9	SN2ND	L5T27,G9,V865	1.41600
2D Partial Core A		1.26661 ±9	MOCEX	L7T21, G9, V1029121, S0.005 L7T21, G11, V1029121, S0.005 L7T15, G9, V610753, S0.01	1.26643 1.26578 1.26735
2D Partial Core B	CR In	1.01919 ±12	MOCEX	L7T15, G11, V924721, S0.01	1.02097
	CR Out	1.03301 ±13		L7T15, G11, V924721, S0.01	1.03456
	CR worth	1312			1286
2D Full Core	CR In	1.01946 ±12	MOCEX	L7T15, G9, V938143, S0.05	1.01565
	CR Out	1.03398 ±12		L7T15, G9, V938143, S0.05	1.03002
	CR worth	1377			1374
3D Single Pin		1.37534 ±10	MOCEX	L7T15, G9, V1537, S0.05	1.37391

* LxTy: Legendre-Tchevychev, G: group, V: vertices, S: ray spacing in cm



(a) 2D Full Core Mesh, (b) 2D Quarter Core Mesh, (c) Zoomed in 2D Quarter Core Mesh,

d) 3D Full Core Mesh, (e) RCF Axial Geometry

Figure 41. 2D and 3D Full Core Meshes of RCF.

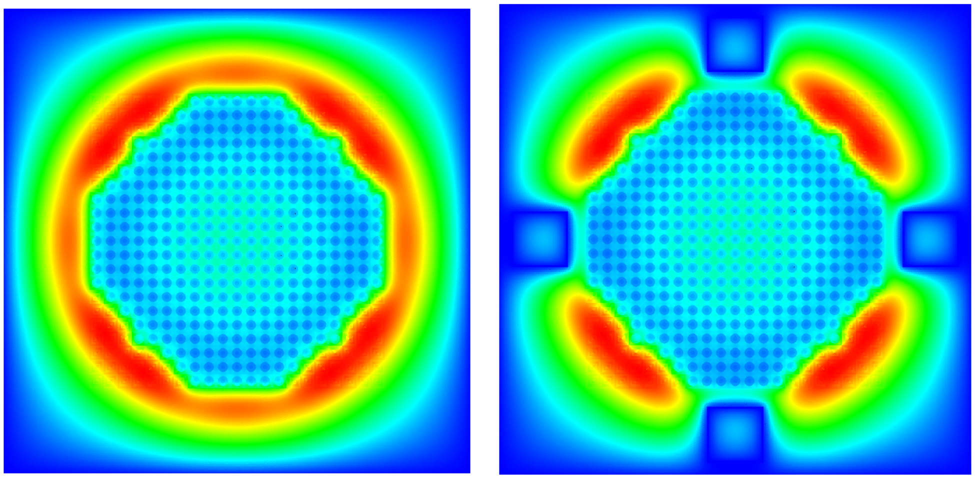


Figure 42. 2D Partial Core of RCF with Control Rods Fast (left) and Thermal Fluxes (right).

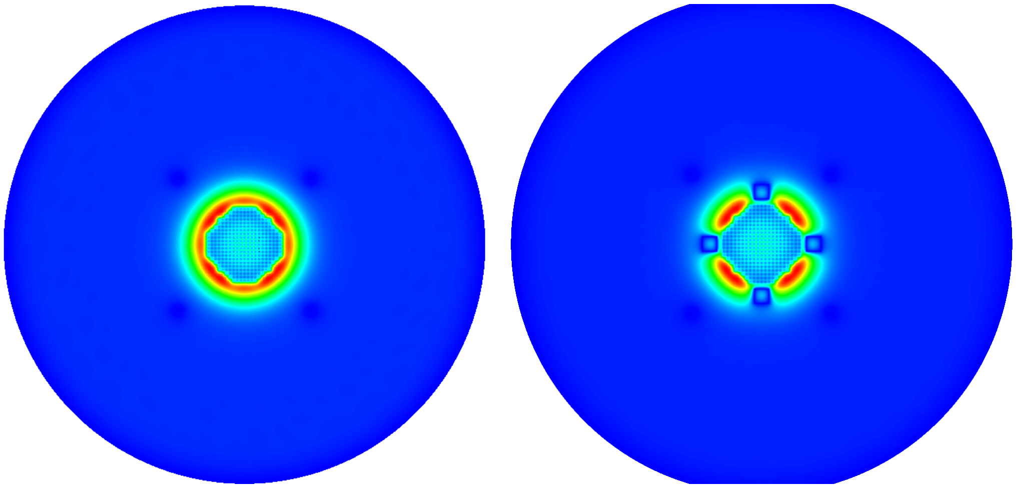


Figure 43. 2D Full Core of RCF with Control Rods Fast (left) and Thermal Fluxes (right).

4. User Support

In this section we discuss efforts relating to user support, training, and documentation. Main activities included a new manual release for PROTEUS-SN, training sessions, and development of promotional materials.

4.1 PROTEUS Release and SHARP Training

The standalone version of PROTEUS version 3.0.0 was released in March 2016, along with the updated PROTEUS-SN User Manual version 3.0.0. The standalone version includes SN2ND, MOCFE, and Cross Section API (CSAPI) so that users can have a choice between SN and MOC solvers depending upon users' needs. The MOCEX solver, which is to be used for 3D simulations rather than MOCFE, was not included in the release since it was under verification at the time of release.

The PROTEUS-SN user training presentations (Methodology and Usage) were also updated. In March 2016, a PROTEUS-SN training session was given as part of a larger SHARP Toolkit Workshop held for potential users of SHARP. Tutorials on Nek5000, PROTEUS-SN, and the SHARP coupled toolkit were given to 11 students and 2 additional Argonne staff members. The PROTEUS-SN tutorial included all of the required information needed to perform basic simulations as well as information its use within the SHARP multiphysics toolkit. Additionally, a similar training session was given to summer interns working in the Neutronics Methods and Codes section and to NEUP university collaborators (University of Michigan, University of Massachusetts-Lowell). By providing training on the codes, these potential users may be more likely to return to their universities and recommend the code to their colleagues for future work.

4.2 Promotional Materials

In order to increase publicity of the PROTEUS package, a professional flyer was created to advertise PROTEUS as shown in Figure 44. The flyer and an addendum describing current research were distributed at the PHYSOR 2016 conference in Sun Valley, Idaho at the Industry Exhibit.

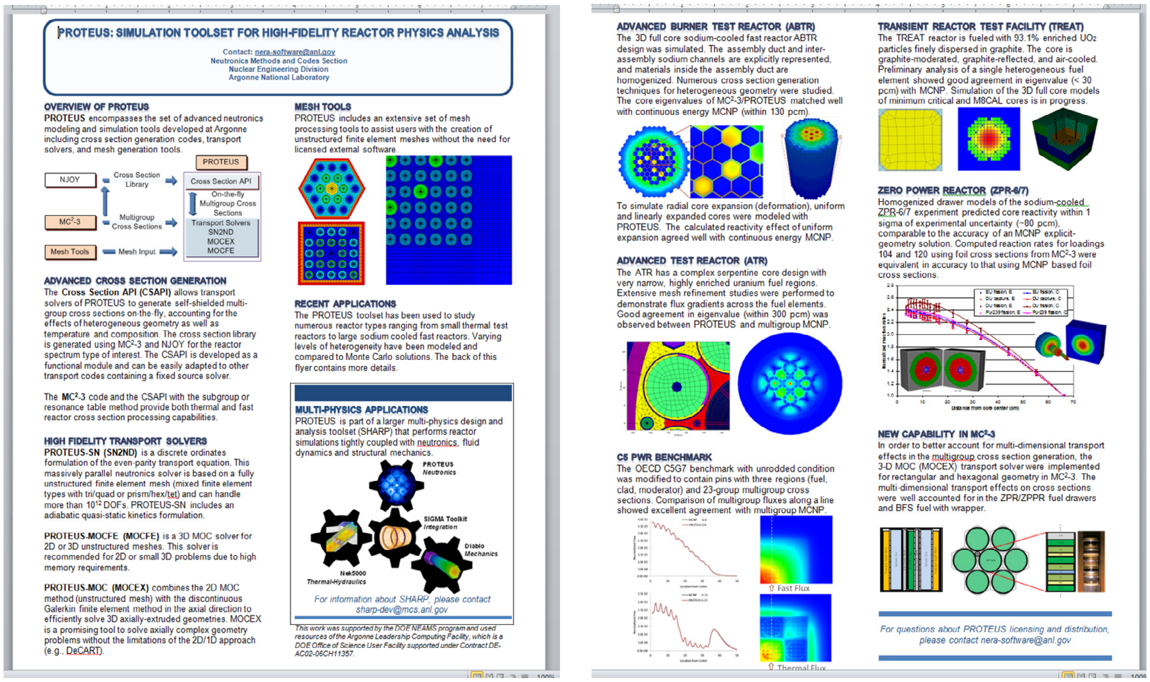
Additionally, a poster describing the SHARP toolkit as well as a poster describing recent advancements with PROTEUS was presented at the Margaret Butler Celebration of Women in Scientific Computing and Nuclear Engineering at Argonne on June 6, 2016.

4.3 PROTEUS and SHARP Support

Since the SHARP release in March 2016, several users have requested to download PROTEUS-SN for both standalone and coupled calculations. Therefore user support needs for both standalone and coupled calculations increased this year due to more users. Some examples of user support provided this year are included here:

- ANL-NE staff ran ASTRID analysis on Blues and BG/Q using PROTEUS-SN.

- ORNL staff are developing the Warthog application (PROTEUS-SN and Bison coupling) and also implementing ORIGEN depletion in PROTEUS-SN.
- NEUP collaborators at different universities are using/developing PROTEUS-SN both in standalone mode and as part of the SHARP toolkit.
- ANL-MCS staff ported SHARP to different computer architectures.
- Several users established SHARP accounts in order to download PROTEUS-SN.



PROTEUS: SIMULATION TOOLSET FOR HIGH-FIDELITY REACTOR PHYSICS ANALYSIS

Contact: nera-software@anl.gov
Neutronics Methods and Codes Section
Nuclear Engineering Division
Argonne National Laboratory

OVERVIEW OF PROTEUS

PROTEUS encompasses the set of advanced neutronics modeling and simulation tools developed at Argonne including cross section generation codes, transport solvers, and mesh generation tools.

NJOY
MC²-3
Mesh Tools
Cross Section Library
Multigroup Cross Sections
Mesh Input
PROTEUS
Cross Section API
On-the-fly Multigroup Cross Sections
Transport Solvers
SN2ND
MOCEX
MOCFE

ADVANCED CROSS SECTION GENERATION

The **Cross Section API (CSAPI)** allows transport solvers of PROTEUS to generate self-shielded multi-group cross sections on-the-fly, accounting for the effects of heterogeneous geometry as well as temperature and composition. The cross section library is generated using MC²-3 and NJOY for the reactor spectrum type of interest. The CSAPI is developed as a functional module and can be easily adapted to other transport codes containing a fixed source solver.

The **MC²-3** code and the CSAPI with the subgroup or resonance table method provide both thermal and fast reactor cross section processing capabilities.

HIGH FIDELITY TRANSPORT SOLVERS

PROTEUS-SN (SN2ND) is a discrete ordinates formulation of the even-parity transport equation. This massively parallel neutronics solver is based on a fully unstructured finite element mesh (mixed finite element types with tri/quad or prism/hex/tet) and can handle more than 10^{12} DOFs. PROTEUS-SN includes an adiabatic quasi-static kinetics formulation.

PROTEUS-MOCFE (MOCFE) is a 3D MOC solver for 2D or 3D unstructured meshes. This solver is recommended for 2D or small 3D problems due to high memory requirements.

PROTEUS-MOC (MOCEX) combines the 2D MOC method (unstructured mesh) with the discontinuous Galerkin finite element method in the axial direction to efficiently solve 3D axially-extruded geometries. MOCEX is a promising tool to solve axially complex geometry problems without the limitations of the 2D/1D approach (e.g., DeCART).

MESH TOOLS

PROTEUS includes an extensive set of mesh processing tools to assist users with the creation of unstructured finite element meshes without the need for licensed external software.

RECENT APPLICATIONS

The PROTEUS toolset has been used to study numerous reactor types ranging from small thermal test reactors to large sodium cooled fast reactors. Varying levels of heterogeneity have been modeled and compared to Monte Carlo solutions. The back of this flyer contains more details.

MULTI-PHYSICS APPLICATIONS

PROTEUS is part of a larger multi-physics design and analysis toolset (SHARP) that performs reactor simulations tightly coupled with neutronics, fluid dynamics and structural mechanics.

For information about SHARP, please contact sharp-dev@mcs.anl.gov

This work was supported by the DOE NEAMS program and used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

ADVANCED BURNER TEST REACTOR (ABTR)

The 3D full core sodium-cooled fast reactor ABTR design was simulated. The assembly duct and inter-assembly sodium channels are explicitly represented, and materials inside the assembly duct are homogenized. Numerous cross section generation techniques for heterogeneous geometry were studied. The core eigenvalues of MC²-3/PROTEUS matched well with continuous energy MCNP (within 130 pcm).

To simulate radial core expansion (deformation), uniform and linearly expanded cores were modeled with PROTEUS. The calculated reactivity effect of uniform expansion agreed well with continuous energy MCNP.

ADVANCED TEST REACTOR (ATR)

The ATR has a complex serpentine core design with very narrow, highly enriched uranium fuel regions. Extensive mesh refinement studies were performed to demonstrate flux gradients across the fuel elements. Good agreement in eigenvalue (within 300 pcm) was observed between PROTEUS and multigroup MCNP.

C5 PWR BENCHMARK

The OECD C5G7 benchmark with unrodded condition was modified to contain pins with three regions (fuel, clad, moderator) and 23-group multigroup cross sections. Comparison of multigroup fluxes along a line showed excellent agreement with multigroup MCNP.

TRANSIENT REACTOR TEST FACILITY (TREAT)

The TREAT reactor is fueled with 93.1% enriched UO_2 particles finely dispersed in graphite. The core is graphite-moderated, graphite-reflected, and air-cooled. Preliminary analysis of a single heterogeneous fuel element showed good agreement in eigenvalue (< 30 pcm) with MCNP. Simulation of the 3D full core models of minimum critical and M8CAL cores is in progress.

ZERO POWER REACTOR (ZPR-6/7)

Homogenized drawer models of the sodium-cooled ZPR-6/7 experiment predicted core reactivity within 1 sigma of experimental uncertainty (~80 pcm), comparable to the accuracy of an MCNP explicit-geometry solution. Computed reaction rates for loadings 104 and 120 using foil cross sections from MC²-3 were equivalent in accuracy to that using MCNP based foil cross sections.

NEW CAPABILITY IN MC²-3

In order to better account for multi-dimensional transport effects in the multigroup cross section generation, the 3-D MOC (MOCEX) transport solver were implemented for rectangular and hexagonal geometry in MC²-3. The multi-dimensional transport effects on cross sections were well accounted for in the ZPR/ZPPR fuel drawers and BFS fuel with wrapper.

For questions about PROTEUS licensing and distribution, please contact nera-software@anl.gov

Figure 44. PROTEUS Flyer

5. Conclusions

The major FY16 accomplishments for NEAMS neutronics activities included significant improvements, in terms of capability and performance, made to the neutronics tools including the PROTEUS transport solvers, MC^2-3 and alternative cross section generation tools, mesh generation and conversion tools, nightly regression tests, and PERSENT. Importantly, significant progress was made on code verification and verification through the demonstration of flexible thermal expansion on SFR applications and the simulation of two experimental thermal reactors. Crucial support was provided to the neutronics tool users as well as the SHARP release, completing the release of the standalone version of PROTEUS before the SHARP release.

The PROTEUS-SN code was updated to integrate both a kinetics mode and mesh deformation capabilities within multi-physics SHARP simulations. Both of these updates were critical to enhancing SHARP neutronics capabilities. Outside of SHARP, a new capability was added to PROTEUS-SN to analyze deformed mesh configurations and process all material densities automatically at runtime based on geometrical changes and material models specified by the user. When used in coupled mode via SHARP, the material densities are also updated automatically which significantly streamlines analysis of radial core expansion problems. Special treatment of liquid sodium is possible such as backfilling sodium into expanded regions. Homogenization and adjustment of homogenized densities is calculated automatically if sodium backfill is selected due to mesh deformation. The new mesh deformation and material density adjustment capabilities were tested both with multi-physics simulations and standalone simulations. PROTEUS-SN's standalone mesh deformation capability was verified against legacy procedures to compute reactivity worth of deformation schemes for the ABTR design using a series of contrived but somewhat realistic deformation schemes. The PROTEUS-SN computed reactivity worths were more conservative (smaller in magnitude) then the conventionally computed quantities, lending motivation to perform direct simulations using high fidelity codes.

The mesh tool utilities for PROTEUS-SN were expanded this year and now include new tools for applying mesh deformation, merging smaller meshes in non-lattice configurations, and modifying the finite element basis functions of any mesh. These tools streamline and facilitate the mesh generation process for PROTEUS's transport solvers which require finite element meshes to define the geometry.

Research into the PROTEUS-MOCEX methodology continued in this year. Several updates were made to the MOCEX solver of PROTEUS including implementation of a Gauss-Seidel scheme in energy that reduces the computational expense and memory requirements along with correcting the underlying mistakes in the ray tracing algorithm. This work was done simultaneously with updating the 2D MOC capability used within MC^2-3 to the MOCEX methodology. As part of this work, a consistent CMFD acceleration scheme was also developed and verified on a variety of thermal and fast spectrum problems. The acceleration proved stable and effective for all tested cases. This development is a key to getting the PROTEUS-MOCEX code to quickly produce solutions on a large problem and thus make it a usable code for reactor analysis.

Additional work was completed on creating an alternative cross section generation capability using the Serpent and OpenMC Monte Carlo codes. This work provides users with several options for cross section generation, such as MC^2-3, online cross section generation using the cross section API, and the two Monte Carlo codes, so that they can select an appropriate option for their problems of interest.

For code verification and validation efforts, two thermal reactor problems, TREAT and RCF, were modeled and simulated using PROTEUS. Systematic verification tests indicate that PROTEUS-MOCEX solutions are in good agreement with Serpent Monte Carlo solutions. In particular, the neutron streaming effect in the 3D TREAT configurations was accurately estimated using PROTEUS-MOCEX. The good agreement seen so far with PROTEUS-MOCEX indicates that we can move on to validation tests with the same experimental reactors in FY17.

In FY17, we anticipate working on several areas that were highlighted as important goals of NEAMS. First, we need to improve the PROTEUS-MOCEX by adding the CMFD acceleration and cleaning up the software with respect to usability and visualization. This will allow us to simulate the fully heterogeneous problems that PROTEUS-SN was never intended to be used for, and has consistently failed to perform well on. In addition to this, we intend on working on a utility code to apply realistic NUBOW-3D deformation data to a PROTEUS-SN mesh. In this case, perturbations to the fuel assembly hexagonal lattice (i.e. fuel assembly ducts touch) can be modeled which is not possible with homogeneous codes like DIF3D and thus we can explicitly demonstrate the mechanical feedback in fast spectrum systems. We also expect to continue work on the cross section generation approaches discussed in the report.

REFERENCES

1. A. SIEGEL et al, "Software Design of SHARP," Proc. of M&C+SNA 2007, Monterey, CA, April 15-17, 2007.
2. E. R. SHEMON, M. A. SMITH, AND C. H. LEE, "PROTEUS-SN Methodology Manual," ANL/NE-14/5, Argonne National Laboratory, June 30, 2014.
3. C. H. LEE and W. S. YANG, "MC^2-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis," ANL/NE-11-41 Rev.1, Argonne National Laboratory, January 2012.
4. M. A. SMITH, W. S. YANG, A. MOHAMED, and E. E. LEWIS, "Perturbation and Sensitivity Tool Based on the VARIANT Option of DIF3D," ANS Transactions **107**, San Diego, CA, Nov. 11-15, 2012.
5. A. MARIN-LAFLECHE, M. A. SMITH, and C. H. LEE, "PROTEUS-MOC: A 3D Deterministic Solver Incorporating 2D Method of Characteristics," Proc. Of International Conf. on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), Sun Valley, Idaho, May 5-9, 2013.
6. J. LEPPANEN, "Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code," VTT Technical Research Centre of Finland, June 18, 2015.
7. P. K. ROMANO et al., "OpenMC: A Status-of-the-Art Monte Carlo Code for Research and Development," *Ann. Nucl. Energy*, **82**, 90-97, 2015.
8. G. A. FREUND et al., "Design Summary Report on the Transient Reactor Test Facility (TREAT)," ANL-6034, Argonne National Laboratory, 1960.
9. N. THOMPSON et al. "Nuclear engineering education at the RPI Walthousen Reactor Critical Facility," INMM Annual Meeting, Indian Wells, CA, 2015.
10. M. A. SMITH et al., "Benchmark on Deterministic Transport Calculations Without Spatial Homogenization (MOX Fuel Assembly 3-D Extension Case)," NEA/NSC/DOC(2005)16, OECD/NEA, 2005.
11. D. W. LEE and H. G. JOO, "Coarse Mesh Finite Difference Acceleration of Discrete Ordinate Neutron Transport Calculation Employing Discontinuous Finite Element Method," *Nucl. Eng. Technol.*, **46**, pp. 783-796, 2014.
12. "Evaluation Guide for the International Reactor Physics Experiments Evaluation Project (IRPhEP)", NEA/NSC/DOC(2006)1, OECD NEA Nuclear Science Committee, March, 2012.
13. R. E. MACFARLANE and A. C. KAHLER, "Methods for Processing ENDF/B-VII with NJOY," *Nuclear Data Sheet*, **111**, 2739, 2010.
14. J. KATAKURA, T. YOSHIDA, K. OYAMATSU, and T. TACHIBANA, "Estimation of beta- and gamma-ray spectra for JENDL FP decay data file," *J. Nucl. Sci. Technol.*, **38**, 470, 2001.
15. J. KATAKURA, "JENDL FP Decay Data File 2011 and Fission Yields Data File 2011," JAEA-Data/Code 2011-025, Japan Atomic Energy Agency, 2012.
16. T. FEI et al, "Calculation of Photon Heat Generation for EBR-II Using DIF3D/MC^2-3," ANS Winter Meeting, Anaheim, California, November 9-13, 2014.

17. G. PALMIOTTI, E. E. LEWIS, and C. B. CARRICO, "VARIANT : VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," ANL-95/40, Argonne National Laboratory, 1995.

18. L. L. BRIGGS et al, "EBR-II Passive Safety Demonstration Tests Benchmark Analyses – Phase 1," *Trans. Am. Nucl. Soc.*, **111**, 1263, 2014.

19. J. J. GRUDZINSKI and C. GRANDY, "Design and Analysis of the Core Restraint System for a Small Modular Fast Reactor," *Transactions of the 22nd Conference on Structural Mechanics in Reactor Technology (SMiRT-22)*, San Francisco, CA, USA, August 18-23, 2013.

20. T. K. KIM, Personal Communication, Argonne National Laboratory, August 2016.

21. M. G. JARRETT, E. R. SHEMON, C. H. LEE, and M. A. SMITH, "Heterogeneous Multigroup Cross Sections for Fast Reactor Calculations with MC^2-3/PROTEUS," Proceedings of PHYSOR 2016, Sun Valley, ID, May 2016.

22. J. J. GRUDZINSKI, Personal Communication, Argonne National Laboratory, August and September 2016.

23. E. R. SHEMON, J. J. GRUDZINSKI, C. H. LEE, J. W. THOMAS, and Y. Q. YU, "Specification of the Advanced Burner Test Reactor Multi-Physics Coupling Demonstration Problem," ANL/NE-15/43, Argonne National Laboratory, December 21, 2015.

24. H. M. Connaway and C. H. LEE, "Preliminary Analysis of the Transient Reactor Test Facility (TREAT) with PROTEUS," ANL/NE-15/36, Argonne National Laboratory, November, 2015.

Argonne
NATIONAL LABORATORY

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

U.S. DEPARTMENT OF
ENERGY

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC